

---

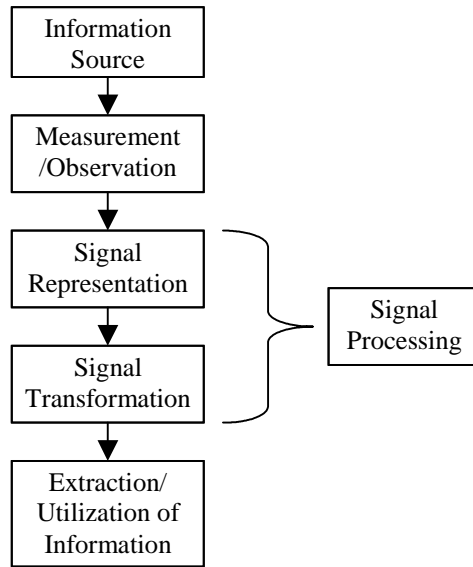
# CHAPTER 5

---

## Digital Signal Processing

One of the most popular ways of characterizing speech is in terms of a *signal* or acoustic waveform. Shown in Figure 5.1 is a representation of the speech signal that ensures that the information content can be easily extracted by human listeners or computers. This is why digital signal processing plays a fundamental role for spoken language processing. We describe here the fundamentals of digital signal processing: digital signals and systems, frequency-domain transforms for both continuous and discrete frequencies, digital filters, the relationship between analog and digital signals, filterbanks, and stochastic processes. In this chapter we set the mathematical foundations of frequency analysis that allow us to develop specific techniques for speech signals in Chapter 6.

The main theme of this chapter is the development of frequency-domain methods computed through the Fourier transform. When we boost the bass knob in our amplifier we are increasing the gain at low frequencies, and when we boost the treble knob we are increasing the gain at high frequencies. Representation of speech signals in the frequency domain is especially useful because the frequency structure of a phoneme is generally unique.

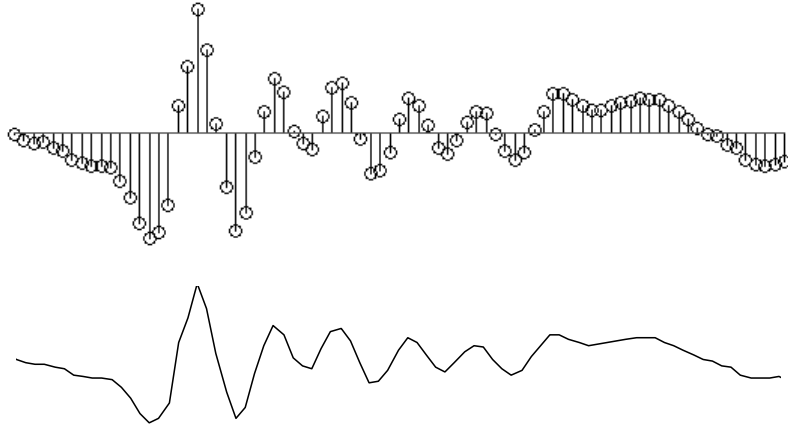


**Figure 5.1** Signal processing is both a representation and a transformation that allows a useful information extraction from a source. The representation and transformation are based on a model of the signal, often parametric, that is convenient for subsequent processing.

## 5.1. DIGITAL SIGNALS AND SYSTEMS

To process speech signals, it is convenient to represent them mathematically as functions of a continuous variable  $t$ , which represents time. Let us define an *analog signal*  $x_a(t)$  as a function varying continuously in time. If we sample the signal  $x$  with a sampling period  $T$  (i.e.,  $t = nT$ ), we can define a discrete-time signal as  $x[n] = x_a(nT)$ , also known as *digital signal*<sup>1</sup>. In this book we use parentheses to describe an analog signal and brackets for digital signals. Furthermore we can define the sampling frequency  $F_s$  as  $F_s = 1/T$ , the inverse of the sampling period  $T$ . For example, for a sampling rate  $F_s = 8\text{kHz}$ , its corresponding sampling period is 125 microseconds. In Section 5.5 it is shown that, under some circumstances, the analog signal  $x_a(t)$  can be recovered exactly from the digital signal  $x[n]$ . Figure 5.2 shows an analog signal and its corresponding digital signal. In subsequent figures, for convenience, we will sometimes plot digital signals as continuous functions.

<sup>1</sup> Actually the term digital signal is defined as a discrete-time signal whose values are represented by integers within a range, whereas a general discrete-time signal would be represented by real numbers. Since the term digital signal is much more commonly used, we will use that term, except when the distinction between them is necessary.



**Figure 5.2** Analog signal and its corresponding digital signal.

The term *Digital Signal Processing* (DSP) refers to methods for manipulating the sequence of numbers  $x[n]$  in a digital computer. The acronym DSP is also used to refer to a *Digital Signal Processor*, i.e., a microprocessor specialized to perform DSP operations.

We start with sinusoidal signals and show they are the fundamental signals for linear systems. We then introduce the concept of convolution and linear time-invariant systems. Other digital signals and nonlinear systems are also introduced.

### 5.1.1. Sinusoidal Signals

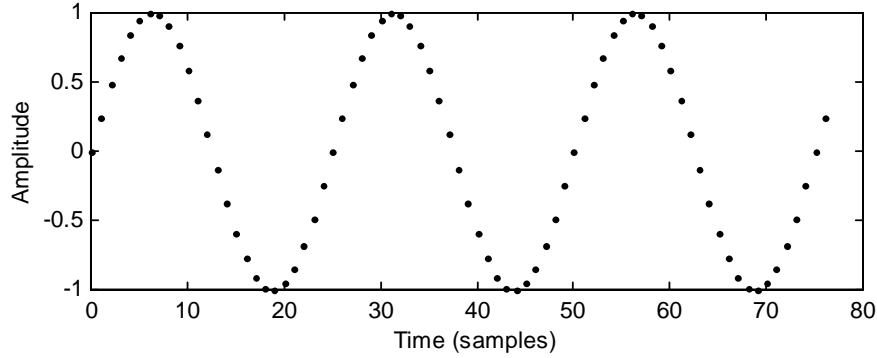
One of the most important signals is the sine wave or *sinusoid*

$$x_0[n] = A_0 \cos(\omega_0 n + \phi_0) \quad (5.1)$$

where  $A_0$  is the sinusoid's amplitude,  $\omega_0$  the angular frequency and  $\phi_0$  the phase. The angle in the trigonometric functions is expressed in radians, so that the angular frequency  $\omega_0$  is related to the normalized linear frequency  $f_0$  by the relation  $\omega_0 = 2\pi f_0$ , and  $0 \leq f_0 \leq 1$ . This signal is *periodic*<sup>2</sup> with period  $T_0 = 1/f_0$ . In Figure 5.3 we can see an example of a sinusoid with frequency  $f_0 = 0.04$ , or a period of  $T_0 = 25$  samples.

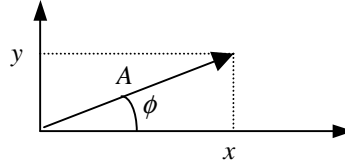
Sinusoids are important because speech signals can be decomposed as sums of sinusoids. When we boost the bass knob in our amplifier we are increasing the gain for sinusoids of low frequencies, and when we boost the treble knob we are increasing the gain for sinusoids of high frequencies.

<sup>2</sup> A signal  $x[n]$  is periodic with period  $N$  if and only if  $x[n] = x[n+N]$ , which requires  $\omega_0 = 2\pi/N$ . This means that the digital signal in Eq. (5.1) is not periodic for all values of  $\omega_0$ , even though its continuous signal counterpart  $x(t) = A_0 \cos(\omega_0 t + \phi_0)$  is periodic for all values of  $\omega_0$  (see Section 5.5).



**Figure 5.3** A digital sinusoid with a period of 25 samples.

What is the sum of two sinusoids  $x_0[n]$  and  $x_1[n]$  of the same frequency  $\omega_0$  but different amplitudes  $A_0$  and  $A_1$ , and phases  $\phi_0$  and  $\phi_1$ ? The answer is another sinusoid of the same frequency but a different amplitude  $A$  and phase  $\phi$ . While this can be computed through trigonometric identities, it is somewhat tedious and not very intuitive. For this reason we introduce another representation based on complex numbers, which proves to be very useful when we study digital filters.



**Figure 5.4** Complex number representation in Cartesian form  $z = x + jy$  and polar form  $z = Ae^{j\phi}$ . Thus  $x = A\cos\phi$  and  $y = A\sin\phi$ .

A complex number  $z$  can be expressed as  $z = x + jy$ , where  $j = \sqrt{-1}$ ,  $x$  is the real part and  $y$  is the imaginary part, with both  $x$  and  $y$  being real numbers. Using Euler's relation, given a real number  $\phi$ , we have

$$e^{j\phi} = \cos\phi + j\sin\phi \quad (5.2)$$

so that the complex number  $z$  can also be expressed in polar form as  $z = Ae^{j\phi}$ , where  $A$  is the amplitude and  $\phi$  is the phase. Both representations can be seen in Figure 5.4, where the real part is shown in the abscissa ( $x$ -axis) and the imaginary part in the ordinate ( $y$ -axis).

Using complex numbers, the sinusoid in Eq. (5.1) can be expressed as the real part of the corresponding complex exponential

$$x_0[n] = A_0 \cos(\omega_0 n + \phi_0) = \text{Re}\{A_0 e^{j(\omega_0 n + \phi_0)}\} \quad (5.3)$$

and thus the sum of two complex exponential signals equals

$$A_0 e^{j(\omega_0 n + \phi_0)} + A_1 e^{j(\omega_0 n + \phi_1)} = e^{j\omega_0 n} (A_0 e^{j\phi_0} + A_1 e^{j\phi_1}) = e^{j\omega_0 n} A e^{j\phi} = A e^{j(\omega_0 n + \phi)} \quad (5.4)$$

Taking the real part in both sides results in

$$A_0 \cos(\omega_0 n + \phi_0) + A_1 \cos(\omega_0 n + \phi_1) = A \cos(\omega_0 n + \phi) \quad (5.5)$$

or in other words, the sum of two sinusoids of the same frequency is another sinusoid of the same frequency.

To compute  $A$  and  $\phi$ , dividing Eq. (5.4) by  $e^{j\omega_0 n}$  leads to a relationship between the amplitude  $A$  and phase  $\phi$ :

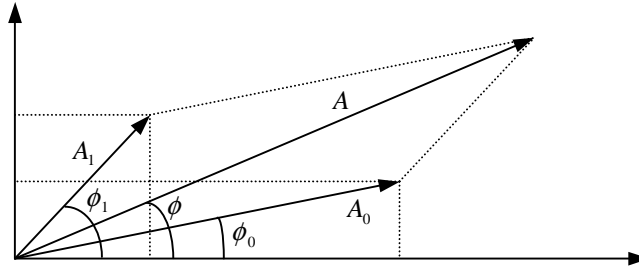
$$A_0 e^{j\phi_0} + A_1 e^{j\phi_1} = A e^{j\phi} \quad (5.6)$$

Equating real and imaginary parts in Eq. (5.6) and dividing them we obtain:

$$\tan \phi = \frac{A_0 \sin \phi_0 + A_1 \sin \phi_1}{A_0 \cos \phi_0 + A_1 \cos \phi_1} \quad (5.7)$$

and adding the squared of real and imaginary parts and using trigonometric identities<sup>3</sup>

$$A^2 = A_0^2 + A_1^2 + 2A_0 A_1 \cos(\phi_0 - \phi_1) \quad (5.8)$$



**Figure 5.5** Geometric representation of the sum of two sinusoids of the same frequency. It follows the complex number representation in Cartesian form of Figure 5.4.

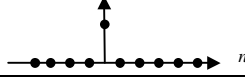
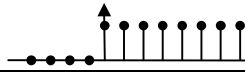
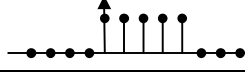
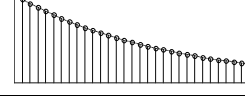
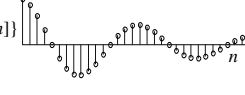
This complex representation of Figure 5.5 lets us analyze and visualize the amplitudes and phases of sinusoids of the same frequency as vectors. The sum of  $N$  sinusoids of the same frequency is another sinusoid of the same frequency that can be obtained by adding the real and imaginary parts of all complex vectors. In Section 5.1.3.3 we show that the output of a linear time-invariant system to a sinusoid is another sinusoid of the same frequency.

<sup>3</sup>  $\sin^2 \phi + \cos^2 \phi = 1$  and  $\cos(a - b) = \cos a \cos b + \sin a \sin b$

### 5.1.2. Other Digital Signals

In the field of digital signal processing there are other signals that repeatedly arise and that are shown in Table 5.1.

**Table 5.1** Some useful digital signals: the Kronecker delta, unit step, rectangular signal, real exponential ( $a < 1$ ) and real part of a complex exponential ( $r < 1$ ).

<i>Kronecker delta, or unit impulse</i>	$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$	
<i>Unit step</i>	$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$	
<i>Rectangular signal</i>	$\text{rect}_N[n] = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$	
<i>Real exponential</i>	$x[n] = a^n u[n]$	
<i>Complex exponential</i>	$x[n] = a^n u[n] = r^n e^{jn\omega_0} u[n]$ $= r^n (\cos n\omega_0 + j \sin n\omega_0) u[n]$	

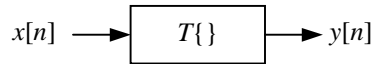
If  $r = 1$  and  $\omega_0 \neq 0$  we have a complex sinusoid as shown in Section 5.1.1. If  $\omega_0 = 0$  we have a real exponential signal, and if  $r < 1$  and  $\omega_0 \neq 0$  we have an exponentially decaying oscillatory sequence, also known as a damped sinusoid.

### 5.1.3. Digital Systems

A digital system is a system that, given an input signal  $x[n]$ , generates an output signal  $y[n]$ :

$$y[n] = T\{x[n]\} \quad (5.9)$$

whose input/output relationship can be seen in Figure 5.6.



**Figure 5.6** Block diagram of a digital system whose input is digital signal  $x[n]$ , and whose output is digital signal  $y[n]$ .

In general, a digital system  $T$  is defined to be linear *iff* (if and only if)

$$T\{a_1 x_1[n] + a_2 x_2[n]\} = a_1 T\{x_1[n]\} + a_2 T\{x_2[n]\} \quad (5.10)$$

for any values of  $a_1$ ,  $a_2$  and any signals  $x_1[n]$  and  $x_2[n]$ .

Here, we study systems according to whether or not they are linear and/or time invariant.

### 5.1.3.1. Linear Time-Invariant Systems

A system is *time-invariant* if given Eq. (5.9), then

$$y[n - n_0] = T\{x[n - n_0]\} \quad (5.11)$$

Linear digital systems of a special type, the so-called *linear time-invariant* (LTI)<sup>4</sup>, are described by

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n] \quad (5.12)$$

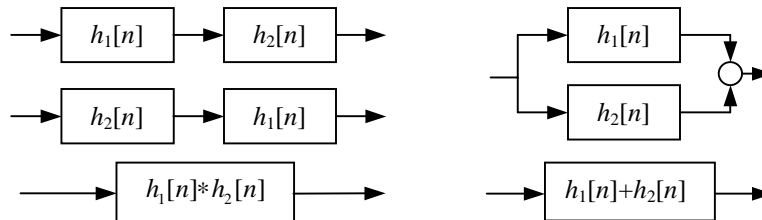
where  $*$  is defined as the *convolution* operator. It is left to the reader to show that the linear system in Eq. (5.12) indeed satisfies Eq. (5.11).

LTI systems are completely characterized by the signal  $h[n]$ , which is known as the system's *impulse response* because it is the output of the system when the input is an impulse  $x[n] = \delta[n]$ . Most of the systems described in this book are LTI systems.

**Table 5.2** Properties of the convolution operator.

Commutative	$x[n] * h[n] = h[n] * x[n]$
Associative	$x[n] * (h_1[n] * h_2[n]) = (x[n] * h_1[n]) * h_2[n] = x[n] * h_1[n] * h_2[n]$
Distributive	$x[n] * (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] * h_2[n]$

The convolution operator is commutative, associative and distributive as shown in Table 5.2 and Figure 5.7.



**Figure 5.7** The block diagrams on the left, representing the commutative property, are equivalent. The block diagrams on the right, representing the distributive property, are also equivalent.

<sup>4</sup> Actually the term linear time-invariant (LTI) systems is typically reserved for continuous or analog systems, and linear shift-invariant system is used for discrete-time signals, but we will use LTI for discrete-time signals too since it is widely used in this context.

### 5.1.3.2. Linear Time-Varying Systems

An interesting type of digital systems is that whose output is a linear combination of the input signal at different times:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]g[n, n-k] \quad (5.13)$$

The digital system in Eq. (5.13) is linear, since it satisfies Eq. (5.10). The Linear Time-Invariant systems of Section 5.1.3.1 are a special case of Eq. (5.13) when  $g[n, n-k] = h[n-k]$ . The systems in Eq. (5.13) are called *linear time-varying* (LTV) systems, because the weighting coefficients can vary with time.

A useful example of such system is the so-called *amplitude modulator*

$$y[n] = x[n]\cos \omega_0 n \quad (5.14)$$

used in AM transmissions. As we show in Chapter 6, speech signals are the output of LTV systems. Since these systems are difficult to analyze, we often approximate them with linear time-invariant systems.

**Table 5.3** Examples of nonlinear systems for speech processing. All of them are memoryless except for the median smoother.

Nonlinear System	Equation
<i>Median Smoother</i> of order $(2N+1)$	$y[n] = \text{median}\{x[n-N], \dots, x[n], \dots, x[n+N]\}$
<i>Full-Wave Rectifier</i>	$y[n] =  x[n] $
<i>Half-Wave Rectifier</i>	$y[n] = \begin{cases} x[n] & x[n] \geq 0 \\ 0 & x[n] < 0 \end{cases}$
<i>Frequency Modulator</i>	$y[n] = A \cos(\omega_0 + \Delta\omega x[n])n$
<i>Hard-Limiter</i>	$y[n] = \begin{cases} A & x[n] \geq A \\ x[n] &  x[n]  < A \\ -A & x[n] \leq -A \end{cases}$
<i>Uniform Quantizer</i> ( $L$ -bit) with $2N = 2^L$ intervals of width $\Delta$	$y[n] = \begin{cases} (N-1/2)\Delta & x[n] \geq (N-1)\Delta \\ (m+1/2)\Delta & m\Delta \leq x[n] < (m+1)\Delta & 0 \leq m < N-1 \\ (-m+1/2)\Delta & -m\Delta \leq x[n] < -(m-1)\Delta & 0 < m < N-1 \\ (-N+1/2)\Delta & x[n] < -(N-1)\Delta \end{cases}$



### 5.1.3.3. Nonlinear Systems

Many *nonlinear* systems do not satisfy Eq. (5.10). Table 5.3 includes a list of typical nonlinear systems used in speech processing. All these nonlinear systems are memoryless, because the output at time  $n$  depends only on the input at time  $n$ , except for the *median smoother* of order  $(2N + 1)$  whose output depends also on the previous and the following  $N$  samples.

## 5.2. CONTINUOUS-FREQUENCY TRANSFORMS

A very useful transform for LTI systems is the Fourier transform, because it uses complex exponentials as its basis functions, and its generalization: the  $z$ -transform. In this section we cover both transforms, which are continuous functions of frequency, and their properties.

### 5.2.1. The Fourier Transform

It is instructive to see what the output of a LTI system with impulse response  $h[n]$  is when the input is a complex exponential. Substituting  $x[n] = e^{j\omega_0 n}$  in Eq. (5.12) and using the commutative property of the convolution we obtain

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0(n-k)} = e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} = e^{j\omega_0 n} H(e^{j\omega_0}) \quad (5.15)$$

which is another complex exponential of the same frequency and amplitude multiplied by the complex quantity  $H(e^{j\omega_0})$  given by

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\omega n} \quad (5.16)$$

Since the output of a LTI system to a complex exponential is another complex exponential, it is said that complex exponentials are *eigensignals* of LTI systems, with the complex quantity  $H(e^{j\omega_0})$  being their *eigenvalue*.

The quantity  $H(e^{j\omega})$  is defined as the *discrete-time Fourier transform* of  $h[n]$ . It is clear from Eq. (5.16) that  $H(e^{j\omega})$  is a periodic function of  $\omega$  with period  $2\pi$ , and therefore we need to keep only one period to fully describe it, typically  $-\pi < \omega < \pi$  (Figure 5.8).

$H(e^{j\omega})$  is a complex function of  $\omega$  which can be expressed in terms of the real and imaginary parts:

$$H(e^{j\omega}) = H_r(e^{j\omega}) + jH_i(e^{j\omega}) \quad (5.17)$$

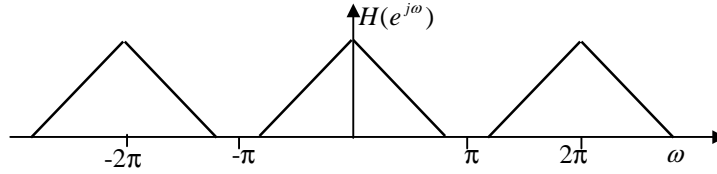
or in terms of the magnitude and phase as

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j \arg[H(e^{j\omega})]} \quad (5.18)$$

Thus if the input to the LTI system is a sinusoid as in Eq. (5.1), the output will be

$$y_0[n] = A_0 |H(e^{j\omega_0})| \cos(\omega_0 n + \phi_0 + \arg\{H(e^{j\omega_0})\}) \quad (5.19)$$

according to Eq. (5.15). Therefore if  $|H(e^{j\omega_0})| > 1$ , the LTI system will amplify that frequency, and likewise it will attenuate, or *filter* it, if  $|H(e^{j\omega_0})| < 1$ . That is one reason why these systems are also called filters. The Fourier transform  $H(e^{j\omega})$  of a filter  $h[n]$  is called the system's *frequency response* or *transfer function*.



**Figure 5.8**  $H(e^{j\omega})$  is a periodic function of  $\omega$ .

The angular frequency  $\omega$  is related to the normalized linear frequency  $f$  by the simple relation  $\omega = 2\pi f$ . We show in Section 5.5 that linear frequency  $f_l$  and normalized frequency  $f$  are related by  $f_l = fF_s$ , where  $F_s$  is the sampling frequency.

The inverse *discrete-time Fourier transform* is defined as

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \quad (5.20)$$

The Fourier transform is invertible, and Eq. (5.16) and (5.20) are transform pairs:

$$\begin{aligned} h[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left( \sum_{m=-\infty}^{\infty} h[m] e^{-j\omega m} \right) e^{j\omega n} d\omega \\ &= \sum_{m=-\infty}^{\infty} h[m] \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega(n-m)} d\omega = \sum_{m=-\infty}^{\infty} h[m] \delta[n-m] = h[n] \end{aligned} \quad (5.21)$$

since

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega(n-m)} d\omega = \delta[n-m] \quad (5.22)$$

A sufficient condition for the existence of the Fourier transform is

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty \quad (5.23)$$

Although we have computed the Fourier transform of the impulse response of a filter  $h[n]$ , Eq. (5.16) and (5.20) can be applied to any signal  $x[n]$ .

### 5.2.2. Z-Transform

The *z-transform* is a generalization of the Fourier transform. The *z-transform* of a digital signal  $h[n]$  is defined as

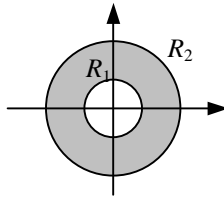
$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n} \quad (5.24)$$

where  $z$  is a complex variable. Indeed, the Fourier transform of  $h[n]$  equals its *z-transform* evaluated at  $z = e^{j\omega}$ . While the Fourier and *z-transforms* are often used interchangeably, we normally use the Fourier transform to plot the filter's frequency response, and the *z-transform* to analyze more general filter characteristics, given its polynomial functional form. We can also use the *z-transform* for unstable filters, which do not have Fourier transforms.

Since Eq. (5.24) is an infinite sum, it is not guaranteed to exist. A sufficient condition for convergence is:

$$\sum_{n=-\infty}^{\infty} |h[n]| |z|^{-n} < \infty \quad (5.25)$$

which is true only for a *region of convergence* (ROC) in the complex *z*-plane  $R_1 < |z| < R_2$  as indicated in Figure 5.9.



**Figure 5.9** Region of convergence of the *z-transform* in the complex plane.

For a signal  $h[n]$  to have a Fourier transform, its *z-transform*  $H(z)$  has to include the unit circle,  $|z| = 1$ , in its convergence region. Therefore, a sufficient condition for the existence of the Fourier transform is given in Eq. (5.23) by applying Eq. (5.25) to the unit circle.

An LTI system is defined to be *causal* if its impulse response is a causal signal, *i.e.*  $h[n] = 0$  for  $n < 0$ . Similarly, a LTI system is *anti-causal* if  $h[n] = 0$  for  $n > 0$ . While all

physical systems are causal, noncausal systems are still useful since causal systems could be decomposed into causal and anti-causal systems.

A system is defined to be *stable* if for every bounded input it produces a bounded output. A necessary and sufficient condition for an LTI system to be stable is

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty \quad (5.26)$$

which means, according to Eq. (5.23), that  $h[n]$  has a Fourier transform, and therefore that its  $z$ -transform includes the unit circle in its region of convergence.

Just as in the case of Fourier transforms, we can use the  $z$ -transform for any signal, not just for a filter's impulse response.

The *inverse  $z$ -transform* is defined as

$$h[n] = \frac{1}{2\pi j} \oint H(z) z^{n-1} dz \quad (5.27)$$

where the integral is performed along a closed contour that is within the *region of convergence*. Eqs. (5.24) and (5.27) plus knowledge of the region of convergence form a transform pair: *i.e.* one can be exactly determined if the other is known. If the integral is performed along the unit circle (*i.e.*, doing the substitution  $z = e^{j\omega}$ ) we obtain Eq. (5.20), the inverse Fourier transform.

### 5.2.3. Z-Transforms of Elementary Functions

In this section we compute the  $z$ -transforms of the signals defined in Table 5.1. The  $z$ -transforms of such signals are summarized in Table 5.4. In particular we compute the  $z$ -transforms of left-sided and right-sided complex exponentials, which are essential to compute the inverse  $z$ -transform of rational polynomials. As we see in Chapter 6, speech signals are often modeled as having  $z$ -transforms that are rational polynomials.

**Table 5.4**  $Z$ -transforms of some useful signals together with their region of convergence.

Signal	$Z$ -Transform	Region of Convergence
$h_1[n] = \delta[n - N]$	$H_1(z) = z^{-N}$	$z \neq 0$
$h_2[n] = u[n] - u[n - N]$	$H_2(z) = \frac{1 - z^{-N}}{1 - z^{-1}}$	$z \neq 0$
$h_3[n] = a^n u[n]$	$H_3(z) = \frac{1}{1 - az^{-1}}$	$ a  <  z $
$h_4[n] = -a^n u[-n - 1]$	$H_4(z) = \frac{1}{1 - az^{-1}}$	$ z  <  a $

### 5.2.3.1. Right-Sided Complex Exponentials

A right-sided complex exponential sequence

$$h_3[n] = a^n u[n] \quad (5.28)$$

has a  $z$ -transform given by

$$H_3(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \lim_{N \rightarrow \infty} \frac{1 - (az^{-1})^{N+1}}{1 - az^{-1}} = \frac{1}{1 - az^{-1}} \quad \text{for } |a| < |z| \quad (5.29)$$

by using the sum of the terms of a geometric sequence and making  $N \rightarrow \infty$ . This region of convergence ( $|a| < |z|$ ) is typical of causal signals (those that are zero for  $n < 0$ ).

When a  $z$ -transform is expressed as the ratio of two polynomials, the roots of the numerator are called *zeros*, and the roots of the denominator are called *poles*. Zeros are the values of  $z$  for which the  $z$ -transform equals 0, and poles are the values of  $z$  for which the  $z$ -transform equals infinity.

$H_3(z)$  has a pole at  $z = a$ , because its value goes to infinity at  $z = a$ . According to Eq. (5.26),  $h_3[n]$  is a stable signal if and only if  $|a| < 1$ , or in other words, if its pole is inside the unit circle. In general, a causal and stable system has all its poles inside the unit circle. As a corollary, a system which has poles outside the unit circle is either noncausal or unstable or both. This is a very important fact, which we exploit throughout the book.

### 5.2.3.2. Left-Sided Complex Exponentials

A left-sided complex exponential sequence

$$h_4[n] = -a^n u[-n-1] \quad (5.30)$$

has a  $z$ -transform given by

$$\begin{aligned} H_4(z) &= -\sum_{n=-\infty}^{-1} a^n z^{-n} = -\sum_{n=1}^{\infty} a^{-n} z^n = 1 - \sum_{n=0}^{\infty} a^{-n} z^n \\ &= 1 - \frac{1}{1 - a^{-1}z} = \frac{-a^{-1}z}{1 - a^{-1}z} = \frac{1}{1 - az^{-1}} \end{aligned} \quad \text{for } |z| < |a| \quad (5.31)$$

This region of convergence ( $|z| < |a|$ ) is typical of noncausal signals (those that are nonzero for  $n < 0$ ). Observe that  $H_3(z)$  and  $H_4(z)$  are functionally identical and only differ in the region of convergence. In general, the region of convergence of a signal that is nonzero for  $-\infty < n < \infty$  is  $R_1 < |z| < R_2$ .

### 5.2.3.3. Inverse Z-Transform of Rational Functions

Integrals in the complex plane such as Eq. (5.27) are not easy to do, but fortunately they are not necessary for the special case of  $H(z)$  being a rational polynomial transform. In this case, partial fraction expansion can be used to decompose the signal into a linear combination of signals like  $h_1[n]$ ,  $h_3[n]$  and  $h_4[n]$  as in Table 5.4.

For example,

$$H_5(z) = \frac{2 + 8z^{-1}}{2 - 5z^{-1} - 3z^{-2}} \quad (5.32)$$

has as roots of its denominator  $z = 3, -1/2$ . Therefore it can be decomposed as

$$H_5(z) = \frac{A}{1 - 3z^{-1}} + \frac{B}{1 + (1/2)z^{-1}} = \frac{(2A + 2B) + (A - 6B)z^{-1}}{2 - 5z^{-1} - 3z^{-2}} \quad (5.33)$$

so that  $A$  and  $B$  are the solution of the following set of linear equations:

$$\begin{aligned} 2A + 2B &= 2 \\ A - 6B &= 8 \end{aligned} \quad (5.34)$$

whose solution is  $A = 2$  and  $B = -1$ , and thus Eq. (5.33) is expressed as

$$H_5(z) = 2 \left( \frac{1}{1 - 3z^{-1}} \right) - \left( \frac{1}{1 + (1/2)z^{-1}} \right) \quad (5.35)$$

However, we cannot compute the inverse  $z$ -transform unless we know the region of convergence. If, for example, we are told that the region of convergence includes the unit circle (necessary for the system to be stable), then the inverse transform of

$$H_4(z) = \frac{1}{1 - 3z^{-1}} \quad (5.36)$$

must have a region of convergence of  $|z| < 3$  according to Table 5.4, and thus be a left-sided complex exponential:

$$h_4[n] = -3^n u[-n - 1] \quad (5.37)$$

and the transform of

$$H_3(z) = \frac{1}{1 + (1/2)z^{-1}} \quad (5.38)$$

must have a region of convergence of  $1/2 < |z|$  according to Table 5.4, and thus be a right-sided complex exponential:

$$h_3[n] = (-1/2)^n u[n] \quad (5.39)$$

so that

$$h_5[n] = -2 \cdot 3^n u[-n-1] - (-1/2)^n u[n] \quad (5.40)$$

While we only showed an example here, the method used generalizes to rational transfer functions with more poles and zeros.

## 5.2.4. Properties of the Z and Fourier Transform

In this section we include a number of properties that are used throughout the book and that can be derived from the definition of Fourier and  $z$ -transforms. Of special interest are the convolution property and Parseval's theorem, which are described below.

### 5.2.4.1. The Convolution Property

The  $z$ -transform of  $y[n]$ , convolution of  $x[n]$  and  $h[n]$ , can be expressed as a function of their  $z$ -transforms:

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y[n] z^{-n} = \sum_{n=-\infty}^{\infty} \left( \sum_{k=-\infty}^{\infty} x[k] h[n-k] \right) z^{-n} \\ &= \sum_{k=-\infty}^{\infty} x[k] \left( \sum_{n=-\infty}^{\infty} h[n-k] z^{-n} \right) = \sum_{k=-\infty}^{\infty} x[k] \left( \sum_{n=-\infty}^{\infty} h[n] z^{-(n+k)} \right) \\ &= \sum_{k=-\infty}^{\infty} x[k] z^{-k} H(z) = X(z) H(z) \end{aligned} \quad (5.41)$$

which is the fundamental property of LTI systems: “The  $z$ -transform of the convolution of two signals is the product of their  $z$ -transforms.” This is also known as the convolution property. The ROC of  $Y(z)$  is now the intersection of the ROCs of  $X(z)$  and  $H(z)$  and cannot be empty for  $Y(z)$  to exist.

Likewise, we can obtain a similar expression for the Fourier transforms:

$$Y(e^{j\omega}) = X(e^{j\omega}) H(e^{j\omega}) \quad (5.42)$$

A dual version of the convolution property can be proven for the product of digital signals:

$$x[n] y[n] \leftrightarrow \frac{1}{2\pi} X(e^{j\omega}) * Y(e^{j\omega}) \quad (5.43)$$

whose transform is the continuous convolution of the transforms with a scale factor. The convolution of functions of continuous variables is defined as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \quad (5.44)$$

Note how this differs from the discrete convolution of Eq. (5.12).

### 5.2.4.2. Power Spectrum and Parseval's Theorem

Let's define the *autocorrelation* of signal  $x[n]$  as

$$R_{xx}[n] = \sum_{m=-\infty}^{\infty} x[m+n]x^*[m] = \sum_{l=-\infty}^{\infty} x[l]x^*[-(n-l)] = x[n] * x^*[-n] \quad (5.45)$$

where the superscript asterisk (\*) means complex conjugate<sup>5</sup> and should not be confused with the convolution operator.

Using the fundamental property of LTI systems in Eq. (5.42) and the symmetry properties in Table 5.5, we can express its Fourier transform  $S_{xx}(\omega)$  as

$$S_{xx}(\omega) = X(\omega)X^*(\omega) = |X(\omega)|^2 \quad (5.46)$$

which is the *power spectrum*. The Fourier transform of the autocorrelation is the power spectrum:

$$R_{xx}[n] \leftrightarrow S_{xx}(\omega) \quad (5.47)$$

or alternatively

$$R_{xx}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{xx}(\omega)e^{j\omega n} d\omega \quad (5.48)$$

If we set  $n = 0$  in Eq. (5.48) and use Eq. (5.45) and (5.46), we obtain

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 d\omega \quad (5.49)$$

which is called *Parseval's theorem* and says that we can compute the signal's energy in the time domain or in the frequency domain.

In Table 5.5 we list, in addition to the convolution property and Parseval's theorem, a number of properties that can be derived from the definition of Fourier and  $z$ -transforms.

## 5.3. DISCRETE-FREQUENCY TRANSFORMS

Here we describe transforms, including the DFT, DCT and FFT, that take our discrete-time signal into a discrete frequency representation. Discrete-frequency transforms are the natural

---

<sup>5</sup> If  $z = x + jy = Ae^{j\phi}$ , its complex conjugate is defined as  $z^* = x - jy = Ae^{-j\phi}$



transform for periodic signals, though we show in Section 5.7 and Chapter 6 how they are also useful for aperiodic signals such as speech.

**Table 5.5** Properties of the Fourier and  $z$ -transforms.

Property	Signal	Fourier Transform	$z$ -Transform
Linearity	$ax_1[n] + bx_2[n]$	$aX_1(e^{j\omega}) + bX_2(e^{j\omega})$	$aX_1(z) + bX_2(z)$
Symmetry	$x[-n]$	$X(e^{-j\omega})$	$X(z^{-1})$
	$x^*[n]$	$X^*(e^{-j\omega})$	$X^*(z^*)$
	$x^*[-n]$	$X^*(e^{j\omega})$	$X^*(1/z^*)$
	$x[n]$ real	$X(e^{j\omega})$ is Hermitian $X(e^{-j\omega}) = X^*(e^{j\omega})$ $ X(e^{j\omega}) $ is even <sup>6</sup> $\text{Re}\{X(e^{j\omega})\}$ is even $\arg\{X(e^{j\omega})\}$ is odd <sup>7</sup> $\text{Im}\{X(e^{j\omega})\}$ is odd	$X(z^*) = X^*(z)$
	Even $\{x[n]\}$	$\text{Re}\{X(e^{j\omega})\}$	
	Odd $\{x[n]\}$	$j \text{Im}\{X(e^{j\omega})\}$	
Time-shifting	$x[n - n_0]$	$X(e^{j\omega})e^{-j\omega n_0}$	$X(z)z^{-n_0}$
Modulation	$x[n]e^{j\omega_0 n}$	$X(e^{j(\omega - \omega_0)})$	$X(e^{-j\omega_0} z)$
	$x[n]z_0^n$		$X(z/z_0)$
Convolution	$x[n] * h[n]$	$X(e^{j\omega})H(e^{j\omega})$	$X(z)H(z)$
	$x[n]y[n]$	$\frac{1}{2\pi} X(e^{j\omega}) * Y(e^{j\omega})$	
Parseval's Theorem	$R_{xx}[n] = \sum_{m=-\infty}^{\infty} x[m+n]x^*[m]$	$S_{xx}(\omega) =  X(\omega) ^2$	$X(z)X^*(1/z^*)$

A discrete transform of a signal  $x[n]$  is another signal defined as

$$X[k] = \mathcal{T}\{x[n]\} \quad (5.50)$$

Linear transforms are special transforms that decompose the input signal  $x[n]$  into a linear combination of other signals:

<sup>6</sup> A function  $f(x)$  is called even if and only if  $f(x) = f(-x)$ .

<sup>7</sup> A function  $f(x)$  is called odd if and only if  $f(x) = -f(-x)$ .

$$x[n] = \sum_{k=-\infty}^{\infty} X[k] \varphi_k[n] \quad (5.51)$$

where  $\varphi_k[n]$  is a set of *orthonormal* functions

$$\langle \varphi_k[n], \varphi_l[n] \rangle = \delta[k - l] \quad (5.52)$$

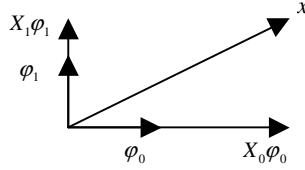
with the inner product defined as

$$\langle \varphi_k[n], \varphi_l[n] \rangle = \sum_{n=-\infty}^{\infty} \varphi_k[n] \varphi_l^*[n] \quad (5.53)$$

With this definition, the coefficients  $X[k]$  are the projection of  $x[n]$  onto  $\varphi_k[n]$ :

$$X[k] = \langle x[n], \varphi_k[n] \rangle \quad (5.54)$$

as illustrated in Figure 5.10.



**Figure 5.10** Orthonormal expansion of a signal  $x[n]$  in a two-dimensional space.

### 5.3.1. The Discrete Fourier Transform (DFT)

If a  $x_N[n]$  signal is periodic with period  $N$  then

$$x_N[n] = x_N[n + N] \quad (5.55)$$

and the signal is uniquely represented by  $N$  consecutive samples. Unfortunately, since Eq. (5.23) is not met, we cannot guarantee the existence of its Fourier transform. The *Discrete Fourier Transform* (DFT) of a periodic signal  $x_N[n]$  is defined as

$$X_N[k] = \sum_{n=0}^{N-1} x_N[n] e^{-j2\pi nk/N} \quad 0 \leq k < N \quad (5.56)$$

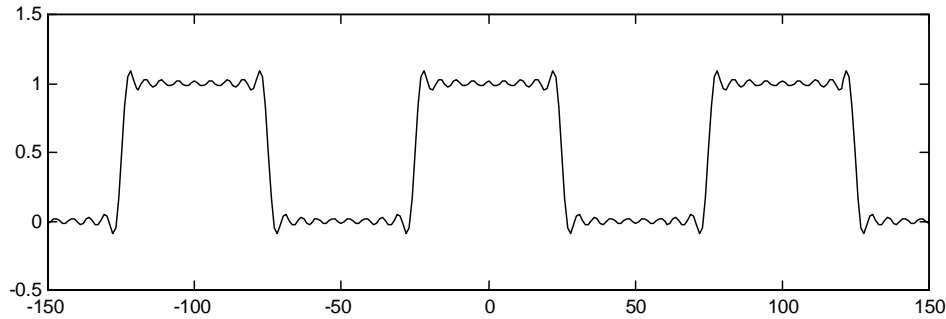
$$x_N[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_N[k] e^{j2\pi nk/N} \quad 0 \leq n < N \quad (5.57)$$

which are transform pairs. Equation (5.57) is also referred to as a *Fourier series* expansion.

In Figure 5.11 we see the approximation of a periodic square signal with period  $N = 100$  as a sum of 19 *harmonic* sinusoids, *i.e.*, we used only the first 19  $X_N[k]$  coefficients in Eq. (5.57).

$$\tilde{x}_N[n] = \frac{1}{N} \sum_{k=-18}^{18} X_N[k] e^{j2\pi nk/N} = \frac{X_N[0]}{N} + \frac{2}{N} \sum_{k=1}^{18} X_N[k] \cos(2\pi nk/N) \quad (5.58)$$

Had we used 100 harmonic sinusoids, the periodic signal would have been reproduced *exactly*. Nonetheless, retaining a smaller number of sinusoids can provide a decent approximation for a periodic signal.



**Figure 5.11** Decomposition of a periodic square signal with period 100 samples as a sum of 19 harmonic sinusoids with frequencies  $\omega_k = 2\pi k/100$ .

### 5.3.2. Fourier Transforms of Periodic Signals

Using the DFT, we now discuss how to compute the Fourier transforms of a complex exponential, an impulse train, and a general periodic signal, since they are signals often used in DSP. We also present a relationship between the continuous-frequency Fourier transform and the discrete Fourier transform.

#### 5.3.2.1. The Complex Exponential

One of the simplest periodic functions is the complex exponential  $x[n] = e^{ja_0 n}$ . Since it has infinite energy, we cannot compute its Fourier transform in its strict sense. Since such signals are so useful, we devise an alternate formulation.

First, let us define the function

$$d_{\Delta}(\omega) = \begin{cases} 1/\Delta & 0 \leq \omega < \Delta \\ 0 & \text{otherwise} \end{cases} \quad (5.59)$$

which has the following property

$$\int_{-\infty}^{\infty} d_{\Delta}(\omega) d\omega = 1 \quad (5.60)$$

for all values of  $\Delta > 0$ .

It is useful to define the continuous delta function  $\delta(\omega)$ , also known as the *Dirac delta*, as

$$\delta(\omega) = \lim_{\Delta \rightarrow 0} d_{\Delta}(\omega) \quad (5.61)$$

which is a *singular function* and can be seen in Figure 5.12. The Dirac delta is a function of a continuous variable and should not be confused with the Kronecker delta, which is a function of a discrete variable.



**Figure 5.12** Representation of the  $\delta(\omega)$  function and its approximation  $d_{\Delta}(\omega)$ .

Using Eqs. (5.59) and (5.61) we can then see that

$$\int_{-\infty}^{\infty} X(\omega) \delta(\omega) d\omega = \lim_{\Delta \rightarrow 0} \int_{-\infty}^{\infty} X(\omega) d_{\Delta}(\omega) d\omega = X(0) \quad (5.62)$$

and similarly

$$\int_{-\infty}^{\infty} X(\omega) \delta(\omega - \omega_0) d\omega = X(\omega_0) \quad (5.63)$$

so that

$$X(\omega) \delta(\omega - \omega_0) = X(\omega_0) \delta(\omega - \omega_0) \quad (5.64)$$

because the integrals on both sides are identical.

Using Eq. (5.63), we see that the convolution of  $X(\omega)$  and  $\delta(\omega - \omega_0)$  is

$$X(\omega) * \delta(\omega - \omega_0) = \int_{-\infty}^{\infty} X(u) \delta(\omega - \omega_0 - u) du = X(\omega - \omega_0) \quad (5.65)$$

For the case of a complex exponential, inserting  $X(\omega) = e^{j\omega n}$  into Eq. (5.63) results in

$$\int_{-\infty}^{\infty} \delta(\omega - \omega_0) e^{j\omega n} d\omega = e^{j\omega_0 n} \quad (5.66)$$

By comparing Eq. (5.66) with (5.20) we can then obtain

$$e^{j\omega_0 n} \leftrightarrow 2\pi\delta(\omega - \omega_0) \quad (5.67)$$

so that the Fourier transform of a complex exponential is an impulse concentrated at frequency  $\omega_0$ .

### 5.3.2.2. The Impulse Train

Since the impulse train

$$p_N[n] = \sum_{k=-\infty}^{\infty} \delta[n - kN] \quad (5.68)$$

is periodic with period  $N$ , it can be expanded in Fourier series according to (5.56) as

$$P_N[k] = 1 \quad (5.69)$$

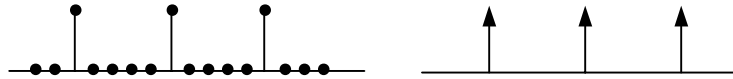
so that using the inverse Fourier series Eq. (5.57),  $p_N[n]$  can alternatively be expressed as

$$p_N[n] = \frac{1}{N} \sum_{k=0}^{N-1} e^{j2\pi kn/N} \quad (5.70)$$

which is an alternate expression to Eq. (5.68) as a sum of complex exponentials. Taking the Fourier transform of Eq. (5.70) and using Eq. (5.67) we obtain

$$P_N(e^{j\omega}) = \frac{2\pi}{N} \sum_{k=0}^{N-1} \delta(\omega - 2\pi k/N) \quad (5.71)$$

which is another impulse train in the frequency domain (See Figure 5.13). The impulse train in the time domain is given in terms of the Kronecker delta, and the impulse train in the frequency domain is given in terms of the Dirac delta.



**Figure 5.13** An impulse train signal and its Fourier transform, which is also an impulse train.

### 5.3.2.3. General Periodic Signals

We now compute the Fourier transform of a general periodic signal using the results of Section 5.3.2.2 and show that, in addition to being periodic, the transform is also discrete. Given a periodic signal  $x_N[n]$  with period  $N$ , we define another signal  $x[n]$ :

$$x[n] = \begin{cases} x_N[n] & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases} \quad (5.72)$$

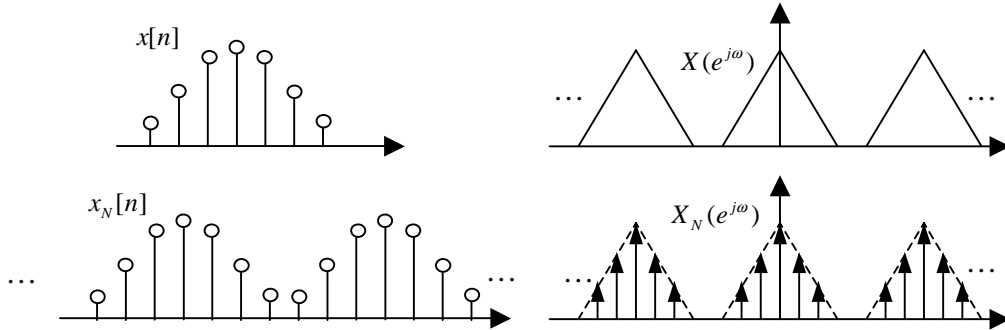
so that

$$x_N[n] = \sum_{k=-\infty}^{\infty} x[n - kN] = x[n] * \sum_{k=-\infty}^{\infty} \delta[n - kN] = x[n] * p_N[n] \quad (5.73)$$

which is the convolution of  $x[n]$  with an impulse train  $p_N[n]$  as in Eq. (5.68). Since  $x[n]$  is of finite length, it has a Fourier transform  $X(e^{j\omega})$ . Using the convolution property  $X_N(e^{j\omega}) = X(e^{j\omega})P_N(e^{j\omega})$ , where  $P_N(e^{j\omega})$  is the Fourier transform of  $p_N[n]$  as given by Eq. (5.71), we obtain another impulse train:

$$X_N(e^{j\omega}) = \frac{2\pi}{N} \sum_{k=-\infty}^{\infty} X(e^{j2\pi k/N}) \delta(\omega - 2\pi k/N) \quad (5.74)$$

Therefore the Fourier transform  $X_N(e^{j\omega})$  of a periodic signal  $x_N[n]$  can be expressed in terms of samples  $\omega_k = 2\pi k/N$ , spaced  $2\pi/N$  apart, of the Fourier transform  $X(e^{j\omega})$  of  $x[n]$ , one period of the signal  $x_N[n]$ . The relationships between  $x[n]$ ,  $x_N[n]$ ,  $X(e^{j\omega})$  and  $X_N(e^{j\omega})$  are shown in Figure 5.14.



**Figure 5.14** Relationships between finite and periodic signals and their Fourier transforms. On one hand,  $x[n]$  is a length  $N$  discrete signal whose transform  $X(e^{j\omega})$  is continuous and periodic with period  $2\pi$ . On the other hand,  $x_N[n]$  is a periodic signal with period  $N$  whose transform  $X_N(e^{j\omega})$  is discrete and periodic.

### 5.3.3. The Fast Fourier Transform (FFT)

There is a family of fast algorithms to compute the DFT, which are called Fast Fourier Transforms (FFT). Direct computation of the DFT from Eq. (5.56) requires  $N^2$  operations, assuming that the trigonometric functions have been pre-computed. The FFT algorithm only requires on the order of  $N \log_2 N$  operations, so it is widely used for speech processing.

### 5.3.3.1. Radix-2 FFT

Let's express the discrete Fourier transform of  $x[n]$

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad 0 \leq k < N \quad (5.75)$$

where we have defined for convenience

$$W_N = e^{-j2\pi/N} \quad (5.76)$$

Equation (5.75) requires  $N^2$  complex multiplies and adds. Now, let's suppose  $N$  is even, and let  $f[n] = x[2n]$  represent the even-indexed samples of  $x[n]$ , and  $g[n] = x[2n+1]$  the odd-indexed samples. We can express Eq. (5.75) as

$$X[k] = \sum_{n=0}^{N/2-1} f[n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} g[n] W_{N/2}^{nk} = F[k] + W_N^k G[k] \quad (5.77)$$

where  $F[k]$  and  $G[k]$  are the  $N/2$  point DFTs of  $f[n]$  and  $g[n]$ , respectively. Since both  $F[k]$  and  $G[k]$  are defined for  $0 \leq k < N/2$ , we need to also evaluate them for  $N/2 \leq k < N$ , which is straightforward, since

$$F[k + N/2] = F[k] \quad (5.78)$$

$$G[k + N/2] = G[k] \quad (5.79)$$

If  $N/2$  is also even, then both  $f[n]$  and  $g[n]$  can be decomposed into sequences of even and odd indexed samples and therefore its DFT can be computed using the same process. Furthermore, if  $N$  is an integer power of 2, this process can be iterated and it can be shown that the number of multiplies and adds is  $N \log_2 N$ , which is a significant saving from  $N^2$ . This is the *decimation-in-time* algorithm and can be seen in Figure 5.15. A dual algorithm called *decimation-in-frequency* can be derived by decomposing the signal into its first  $N/2$  and its last  $N/2$  samples.

### 5.3.3.2. Other FFT Algorithms

Although the radix-2 FFT is the best known algorithm, there are other variants that are faster and are more often used in practice. Among those are the radix-4, radix-8, split-radix and prime-factor algorithm.

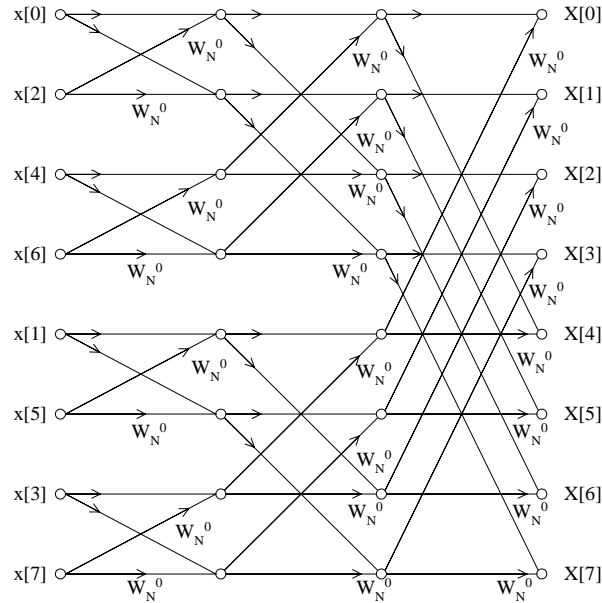
The same process used in the derivation of the radix-2 decimation-in-time algorithm applies if we decompose the sequences into four sequences:  $f_1[n] = x[4n]$ ,  $f_2[n] = x[4n+1]$ ,  $f_3[n] = x[4n+2]$ , and  $f_4[n] = x[4n+3]$ . This is the radix-4 algorithm,

which can be applied when  $N$  is a power of 4, and is generally faster than an equivalent radix-2 algorithm.

Similarly there are radix-8 and radix-16 algorithms for  $N$  being powers of 8 and 16 respectively, which use fewer multiplies and adds. But because of possible additional control logic, it is not obvious that they will be faster, and every algorithm needs to be optimized for a given processor.

There are values of  $N$ , such as  $N = 128$ , for which we cannot use radix-4, radix-8 nor radix-16, so we have to use the less efficient radix-2. A combination of radix-2 and radix-4, called *split-radix* [5], has been shown to have fewer multiplies than both radix-2 and radix-4, and can be applied to  $N$  being a power of 2.

Finally, another possible decomposition is  $N = p_1 p_2 \cdots p_L$  with  $p_i$  being prime numbers. This leads to the *prime-factor algorithm* [2]. While this family of algorithms offers a similar number of operations as the algorithms above, it offers more flexibility in the choice of  $N$ .



**Figure 5.15** Decimation in time radix-2 algorithm for an 8-point FFT.

### 5.3.3.3. FFT Subroutines

Typically, FFT subroutines are computed *in-place* to save memory and have the form

```
fft (float *xr, float *xi, int n)
```

where  $xr$  and  $xi$  are the real and imaginary parts respectively of the input sequence, before calling the subroutine, and the real and imaginary parts of the output transform, after return-



ing from it. C code that implements a decimation-in-time radix-2 FFT of Figure 5.15 is shown in Figure 5.16.

```
void fft2 (float *x, float *y, int n, int m)
{
    int n1, n2, i, j, k, l;
    float xt, yt, c, s;
    double e, a;

    /* Loop through all m stages */
    n2 = n;
    for (k = 0; k < m; k++) {
        n1 = n2;
        n2 = n2 / 2;
        e = PI2 / n1;
        for (j = 0; j < n2; j++) {
            /* Compute Twiddle factors */
            a = j * e;
            c = (float) cos (a);
            s = (float) sin (a);

            /* Do the butterflies */
            for (i = j; i < n; i += n1) {
                l = i + n2;
                xt = x[i] - x[l];
                x[i] = x[i] + x[l];
                yt = y[i] - y[l];
                y[i] = y[i] + y[l];
                x[l] = c * xt + s * yt;
                y[l] = c * yt - s * xt;
            }
        }

        /* Bit reversal: descrambling */
        j = 0;
        for (i = 0; i < n - 1; i++) {
            if (i < j) {
                xt = x[j];
                x[j] = x[i];
                x[i] = xt;
                yt = y[j];
                y[j] = y[i];
                y[i] = yt;
            }
            k = n / 2;
            while (k <= j) {
                j -= k;
                k /= 2;
            }
            j += k;
        }
    }
}
```

**Figure 5.16** C source for a decimation-in-time radix-2 FFT. Before calling the subroutine,  $x$  and  $y$  contain the real and imaginary parts of the input signal respectively. After returning from the subroutine,  $x$  and  $y$  contain the real and imaginary parts of the Fourier transform of the input signal.  $n$  is the length of the FFT and is related to  $m$  by  $n = 2^m$ .

The first part of the subroutine in Figure 5.16 is doing the so-called *butterflies*, which use the trigonometric factors, also called *twiddle factors*. Normally, those twiddle factors are

pre-computed and stored in a table. The second part of the subroutine deals with the fact that the output samples are not linearly ordered (see Figure 5.15), in fact the indexing has the bits reversed, which is why we need to do *bit reversal*, also called *descrambling*.

To compute the inverse FFT an additional routine is not necessary; it can be computed with the subroutine above. To see that, we expand the DFT in Eq. (5.56) into its real and imaginary parts:

$$X_R[k] + jX_I[k] = \sum_{n=0}^{N-1} (x_R[n] + jx_I[n]) e^{-j2\pi nk/N} \quad (5.80)$$

take complex conjugate and multiply by  $j$  to obtain

$$X_I[k] + jX_R[k] = \sum_{n=0}^{N-1} (x_I[n] + jx_R[n]) e^{j2\pi nk/N} \quad (5.81)$$

which has the same functional form as the expanded inverse DFT of Eq. (5.57)

$$x_R[k] + jx_I[k] = \frac{1}{N} \sum_{n=0}^{N-1} (X_R[n] + jX_I[n]) e^{j2\pi nk/N} \quad (5.82)$$

so that the inverse FFT can be computed by calling `fft (xi, xr, n)` other than the  $(1/N)$  factor.

Often the input signal  $x[n]$  is real, so that we know from the symmetry properties of Table 5.5 that its Fourier transform is Hermitian. This symmetry can be used to compute the length- $N$  FFT more efficiently with a length  $(N/2)$  FFT. One way of doing so is to define  $f[n] = x[2n]$  to represent the even-indexed samples of  $x[n]$ , and  $g[n] = x[2n+1]$  the odd-indexed samples. We can then define a length  $(N/2)$  complex signal  $h[n]$  as

$$h[n] = f[n] + jg[n] = x[2n] + jx[2n+1] \quad (5.83)$$

whose DFT is

$$H[k] = F[k] + jG[k] = H_R[k] + jH_I[k] \quad (5.84)$$

Since  $f[n]$  and  $g[n]$  are real, their transforms are Hermitian and thus

$$H^*[-k] = F^*[-k] - jG^*[-k] = F[k] - jG[k] \quad (5.85)$$

Using Eqs. (5.84) and (5.85), we can obtain  $F[k]$  and  $G[k]$  as a function of  $H_R[k]$  and  $H_I[k]$ :

$$F[k] = \frac{H[k] + H^*[-k]}{2} = \left( \frac{H_R[k] + H_R[-k]}{2} \right) + j \left( \frac{H_I[k] - H_I[-k]}{2} \right) \quad (5.86)$$

$$G[k] = \frac{H[k] - H^*[-k]}{2j} = \left( \frac{H_I[k] + H_I[-k]}{2} \right) - j \left( \frac{H_R[k] - H_R[-k]}{2} \right) \quad (5.87)$$

As shown in Eq. (5.77),  $X[k]$  can be obtained as a function of  $F[k]$  and  $G[k]$

$$X[k] = F[k] + G[k]W_N^{-k} \quad (5.88)$$

so that the DFT of the real sequence  $x[n]$  is obtained through Eqs. (5.83), (5.86), (5.87) and (5.88). The computational complexity is a length  $(N/2)$  complex FFT plus  $N$  real multiplies and  $3N$  real adds.

### 5.3.4. Circular Convolution

The convolution of two periodic signals is not defined according to Eq. (5.12). Given two periodic signals  $x_1[n]$  and  $x_2[n]$  with period  $N$ , we define their *circular convolution* as

$$y[n] = x_1[n] \otimes x_2[n] = \sum_{m=0}^{N-1} x_1[m]x_2[n-m] = \sum_{m=\langle N \rangle} x_1[m]x_2[n-m] \quad (5.89)$$

where  $m = \langle N \rangle$  in Eq. (5.89) means that the sum lasts only one period. In fact, the sum could be over any  $N$  consecutive samples, not just the first  $N$ . Moreover,  $y[n]$  is also periodic with period  $N$ . Furthermore, it is left to the reader to show that

$$Y[k] = X_1[k]X_2[k] \quad (5.90)$$

i.e., the DFT of  $y[n]$  is the product of the DFTs of  $x_1[n]$  and  $x_2[n]$ .

An important application of the above result is the computation of a regular convolution using a circular convolution. Let  $x_1[n]$  and  $x_2[n]$  be two signals such that  $x_1[n] = 0$  outside  $0 \leq n < N_1$ , and  $x_2[n] = 0$  outside  $0 \leq n < N_2$ . We know that their regular convolution  $y[n] = x_1[n] * x_2[n]$  is zero outside  $0 \leq N_1 + N_2 - 1$ . If we choose an integer  $N$  such that  $N \geq N_1 + N_2 - 1$ , we can define two periodic signals  $\tilde{x}_1[n]$  and  $\tilde{x}_2[n]$  with period  $N$  such that

$$\tilde{x}_1[n] = \begin{cases} x_1[n] & 0 \leq n < N_1 \\ 0 & N_1 \leq n < N \end{cases} \quad (5.91)$$

$$\tilde{x}_2[n] = \begin{cases} x_2[n] & 0 \leq n < N_2 \\ 0 & N_2 \leq n < N \end{cases} \quad (5.92)$$

where  $x_1[n]$  and  $x_2[n]$  have been *zero padded*. It can be shown that the circular convolution  $\tilde{y}[n] = \tilde{x}_1[n] \otimes \tilde{x}_2[n]$  is identical to  $y[n]$  for  $0 \leq n < N$ , which means that  $y[n]$  can be ob-

tained as the inverse DFT of  $\tilde{Y}[k] = \tilde{X}_1[k]\tilde{X}_2[k]$ . This method of computing the regular convolution of two signals is more efficient than the direct calculation when  $N$  is large. While the crossover point will depend on the particular implementations of the FFT and convolution, as well as the processor, in practice this has been found beneficial for  $N \geq 1024$ .

### 5.3.5. The Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a widely used for speech processing. It has several definitions. The DCT-II  $C[k]$  of a real signal  $x[n]$  is defined by:

$$C[k] = \sum_{n=0}^{N-1} x[n] \cos(\pi k(n+1/2)/N) \quad \text{for } 0 \leq k < N \quad (5.93)$$

with its inverse given by

$$x[n] = \frac{1}{N} \left\{ C[0] + 2 \sum_{k=1}^{N-1} C[k] \cos(\pi k(n+1/2)/N) \right\} \quad \text{for } 0 \leq n < N \quad (5.94)$$

The DCT-II can be derived from the DFT by assuming  $x[n]$  is a real periodic sequence with period  $2N$  and with an even symmetry  $x[n] = x[2N-1-n]$ . It is left to the reader to show, that  $X[k]$  and  $C[k]$  are related by

$$X[k] = 2e^{j\pi k/2N} C[k] \quad \text{for } 0 \leq k < N \quad (5.95)$$

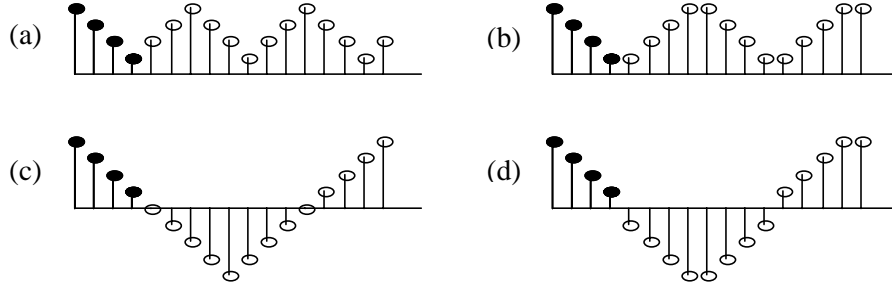
$$X[2N-k] = 2e^{-j\pi k/2N} C[k] \quad \text{for } 0 \leq k < N \quad (5.96)$$

It is left to the reader to prove Eq. (5.94) is indeed the inverse transform using Eqs. (5.57), (5.95), and (5.96). Other versions of the DCT-II have been defined that differ on the normalization constants but are otherwise the same.

There are eight different ways to extend an  $N$ -point sequence and make it both periodic and even, such that can be uniquely recovered. The DCT-II is just one of the ways, with three others being shown in Figure 5.17.

The DCT-II is the most often used discrete cosine transform because of its *energy compaction*, which results in its coefficients being more concentrated at low indices than the DFT. This property allows us to approximate the signal with fewer coefficients [10].

From Eq. (5.95) and (5.96) we see that the DCT-II of a real sequence can be computed with a length- $2N$  FFT of a real and even sequence, which in turn can be computed with a length  $(N/2)$  complex FFT and some additional computations. Other fast algorithms have been derived to compute the DCT directly [15], using the principles described in Section 5.3.3.1. Two-dimensional transforms can also be used for image processing.



**Figure 5.17** Four ways to extend a four-point sequence  $x[n]$  to make it both periodic and have even symmetry. The figures in (a), (b), (c) and (d) correspond to the DCT-I, DCT-II, DCT-III and DCT-IV respectively.

## 5.4. DIGITAL FILTERS AND WINDOWS

We describe here the fundamentals of digital filter design and study *finite-impulse response* (FIR) and *infinite-impulse response* (IIR) filters, which are special types of linear time-invariant digital filters. We establish the time-frequency duality and study the ideal low-pass filter (frequency limited) and its dual window functions (time limited). These transforms are applied to stochastic processes.

### 5.4.1. The Ideal Low-Pass Filter

It is useful to find an impulse response  $h[n]$  whose Fourier transform is

$$H(e^{j\omega}) = \begin{cases} 1 & |\omega| < \omega_0 \\ 0 & \omega_0 < |\omega| < \pi \end{cases} \quad (5.97)$$

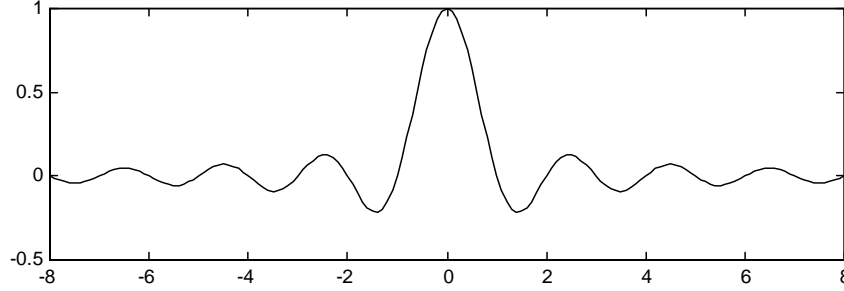
which is the ideal *low-pass* filter because it lets all frequencies below  $\omega_0$  pass through unaffected and completely blocks frequencies above  $\omega_0$ . Using the definition of Fourier transform, we obtain

$$h[n] = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} e^{j\omega n} d\omega = \frac{(e^{j\omega_0 n} - e^{-j\omega_0 n})}{2\pi j n} = \frac{\sin \omega_0 n}{\pi n} = \left(\frac{\omega_0}{\pi}\right) \text{sinc}(\omega_0 n) \quad (5.98)$$

where we have defined the so-called sinc function as

$$\text{sinc}(x) = \frac{\sin \pi x}{\pi x} \quad (5.99)$$

which is a real and even function of  $x$  and is plotted in Figure 5.18. Note that the sinc function is 0 when  $x$  is a nonzero integer.



**Figure 5.18** A sinc function, which is the impulse response of the ideal low-pass filter with a scale factor.

Thus, an ideal low-pass filter is noncausal since it has an impulse response with an infinite number of nonzero coefficients.

## 5.4.2. Window Functions

Window functions are signals that are concentrated in time, often of limited duration. While window functions such as *triangular*, *Kaiser*, *Barlett*, and *prolate spheroidal* occasionally appear in digital speech processing systems, the rectangular, Hanning, and Hamming are the most widely used. Window functions are also concentrated in low frequencies. These window functions are useful in digital filter design and all throughout Chapter 6.

### 5.4.2.1. The Rectangular Window

The *rectangular* window is defined as

$$h_{\pi}[n] = u[n] - u[n - N] \quad (5.100)$$

and we refer to it often in this book. Its  $z$ -transform is given by

$$H_{\pi}(z) = \sum_{n=0}^{N-1} z^{-n} \quad (5.101)$$

which results in a polynomial of order  $(N - 1)$ . Multiplying both sides of Eq. (5.101) by  $z^{-1}$ , we obtain

$$z^{-1}H_{\pi}(z) = \sum_{n=1}^N z^{-n} = H_{\pi}(z) - 1 + z^{-N} \quad (5.102)$$

and therefore the sum of the terms of a geometric series can also be expressed as

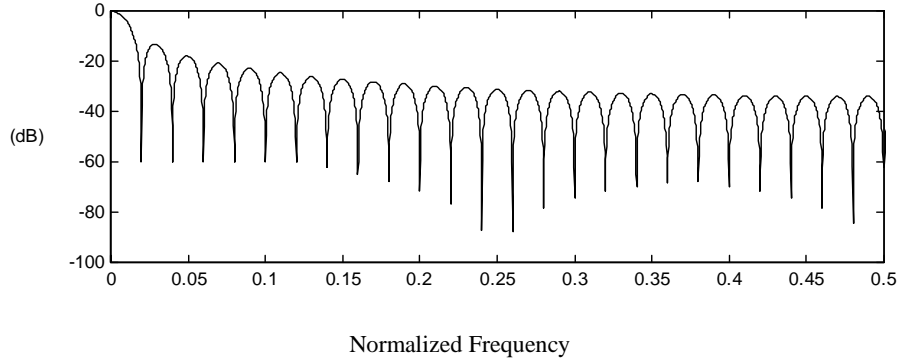
$$H_{\pi}(z) = \frac{1 - z^{-N}}{1 - z^{-1}} \quad (5.103)$$

Although  $z = 1$  appears to be a pole in Eq. (5.103), it actually isn't because it is canceled by a zero at  $z = 1$ . Since  $h_{\pi}[n]$  has finite length, Eq. (5.25) must be satisfied for  $z \neq 0$ , so the region of convergence is everywhere but at  $z = 0$ . Moreover, all finite-length sequences have a region of convergence that is the complete  $z$ -plane except for possibly  $z = 0$ .

The Fourier transform of the rectangular window is, using Eq. (5.103):

$$\begin{aligned} H_{\pi}(e^{j\omega}) &= \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} = \frac{(e^{j\omega N/2} - e^{-j\omega N/2})e^{-j\omega N/2}}{(e^{j\omega/2} - e^{-j\omega/2})e^{-j\omega/2}} \\ &= \frac{\sin \omega N / 2}{\sin \omega / 2} e^{-j\omega(N-1)/2} = A(\omega) e^{-j\omega(N-1)/2} \end{aligned} \quad (5.104)$$

where  $A(\omega)$  is real and even. The function  $A(\omega)$ , plotted in Figure 5.19 in dB,<sup>8</sup> is 0 for  $\omega_k = 2\pi k / N$  with  $k \neq \{0, \pm N, \pm 2N, \dots\}$ , and is the discrete-time equivalent of the sinc function.



**Figure 5.19** Frequency response (magnitude in dB) of the rectangular window with  $N = 50$ , which is a digital sinc function.

#### 5.4.2.2. The Generalized Hamming Window

The *generalized Hamming window* is defined as

<sup>8</sup> An energy value  $E$  is expressed in decibels (dB) as  $\bar{E} = 10 \log_{10} E$ . If the energy value is  $2E$ , it is therefore 3dB higher. Logarithmic measurements like dB are useful because they correlate well with how the human auditory system perceives volume.

$$h_h[n] = \begin{cases} (1-\alpha) - \alpha \cos(2\pi n/N) & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases} \quad (5.105)$$

and can be expressed in terms of the rectangular window in Eq. (5.100) as

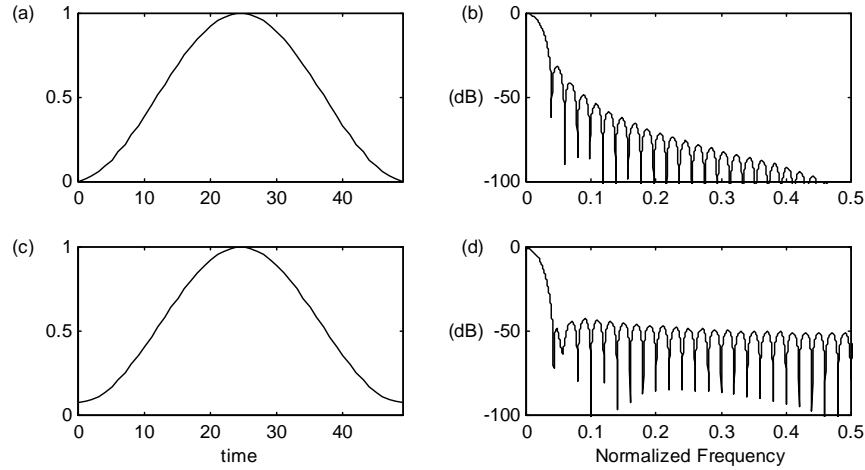
$$h_h[n] = (1-\alpha)h_\pi[n] - \alpha h_\pi[n] \cos(2\pi n/N) \quad (5.106)$$

whose transform is

$$H_h(e^{j\omega}) = (1-\alpha)H_\pi(e^{j\omega}) - (\alpha/2)H_\pi(e^{j(\omega-2\pi/N)}) - (\alpha/2)H_\pi(e^{j(\omega+2\pi/N)}) \quad (5.107)$$

after using the modulation property in Table 5.5. When  $\alpha = 0.5$  the window is known as the *Hanning* window, whereas for  $\alpha = 0.46$  it is the *Hamming* window. Hanning and Hamming windows and their magnitude frequency responses are plotted in Figure 5.20.

The main lobe of both Hamming and Hanning is twice as wide as that of the rectangular window, but the attenuation is much greater than that of the rectangular window. The secondary lobe of the Hanning window is 31 dB below the main lobe, whereas for the Hamming window it is 44 dB below. On the other hand, the attenuation of the Hanning window decays with frequency quite rapidly, which is not the case for the Hamming window, whose attenuation stays approximately constant for all frequencies.



**Figure 5.20** (a) Hanning window and (b) the magnitude of its frequency response in dB; (c) Hamming window and (d) the magnitude of its frequency response in dB for  $N = 50$ .

### 5.4.3. FIR Filters

From a practical point of view, it is useful to consider LTI filters whose impulse responses have a limited number of nonzero coefficients:



$$h[n] = \begin{cases} b_n & 0 \leq n \leq M \\ 0 & \text{otherwise} \end{cases} \quad (5.108)$$

These types of LTI filters are called *finite-impulse response* (FIR) filters. The input/output relationship in this case is

$$y[n] = \sum_{r=0}^M b_r x[n-r] \quad (5.109)$$

The  $z$ -transform of  $x[n-r]$  is

$$\sum_{n=-\infty}^{\infty} x[n-r] z^{-n} = \sum_{n=-\infty}^{\infty} x[n] z^{-(n+r)} = z^{-r} X(z) \quad (5.110)$$

Therefore, given that the  $z$ -transform is linear,  $H(z)$  is

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{r=0}^M b_r z^{-r} = A z^{-L} \prod_{r=1}^{M-L} (1 - c_r z^{-1}) \quad (5.111)$$

whose region of convergence is the whole  $z$ -plane except for possibly  $z = 0$ . Since  $\sum_{r=0}^M |b_r|$  is finite, FIR systems are always stable, which makes them very attractive. Several special types of FIR filters will be analyzed below: linear-phase, first-order and low-pass FIR filters.

#### 5.4.3.1. Linear-Phase FIR Filters

Linear-phase filters are important because, other than a delay, the phase of the signal is unchanged. Only the magnitude is affected. Therefore, the temporal properties of the input signal are preserved. In this section we show that linear-phase FIR filters can be built if the filter exhibits symmetry.

Let's explore the particular case of  $h[n]$  real,  $M = 2L$ , an even number, and  $h[n] = h[M-n]$  (called a *Type-I* filter). In this case

$$\begin{aligned} H(e^{j\omega}) &= \sum_{n=0}^M h[n] e^{-j\omega n} = h[L] e^{-j\omega L} + \sum_{n=0}^{L-1} (h[n] e^{-j\omega n} + h[M-n] e^{-j\omega(2L-n)}) \\ &= h[L] e^{-j\omega L} + \sum_{n=0}^{L-1} h[n] (e^{-j\omega(n-L)} + e^{j\omega(n-L)}) e^{-j\omega L} \\ &= \left( h[L] + \sum_{n=1}^L 2h[n+L] \cos(\omega n) \right) e^{-j\omega L} = A(\omega) e^{-j\omega L} \end{aligned} \quad (5.112)$$

where  $A(\omega)$  is a real and even function of  $\omega$ , since the cosine is an even function, and  $A(\omega)$  is a linear combination of cosines. Furthermore, we see that the phase

$\arg\{H(e^{j\omega})\} = L\omega$ , which is a linear function of  $\omega$ , and therefore  $h[n]$  is called a *linear-phase* system. It can be shown that if  $h[n] = -h[M-n]$ , we also get a linear phase system but  $A(\omega)$  this time is a pure imaginary and odd function (Type III filter). It is left to the reader to show that in the case of  $M$  being odd the system is still linear phase (Types II and IV filters). Moreover,  $h[n]$  doesn't have to be real and:

$$h[n] = \pm h^*[M-n] \quad (5.113)$$

is a sufficient condition for  $h[n]$  to be linear phase.

#### 5.4.3.2. First-Order FIR Filters

A special case of FIR filters is the first-order filter:

$$y[n] = x[n] + \alpha x[n-1] \quad (5.114)$$

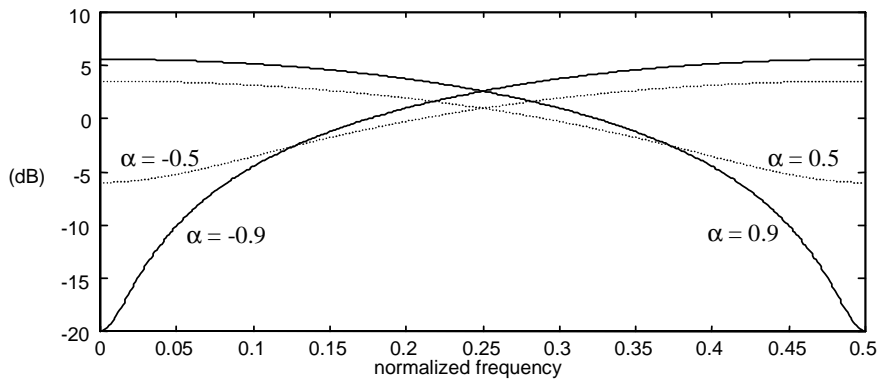
for real values of  $\alpha$ , which, unless  $\alpha = 1$ , is not linear phase. Its  $z$ -transform is

$$H(z) = 1 + \alpha z^{-1} \quad (5.115)$$

It is of interest to analyze the magnitude and phase of its frequency response

$$\begin{aligned} |H(e^{j\omega})|^2 &= |1 + \alpha(\cos \omega - j \sin \omega)|^2 \\ &= (1 + \alpha \cos \omega)^2 + (\alpha \sin \omega)^2 = 1 + \alpha^2 + 2\alpha \cos \omega \end{aligned} \quad (5.116)$$

$$\theta(e^{j\omega}) = -\arctan\left(\frac{\alpha \sin \omega}{1 + \alpha \cos \omega}\right) \quad (5.117)$$



**Figure 5.21** Frequency response of the first order FIR filter for various values of  $\alpha$ .

It is customary to display the magnitude response in decibels (dB):

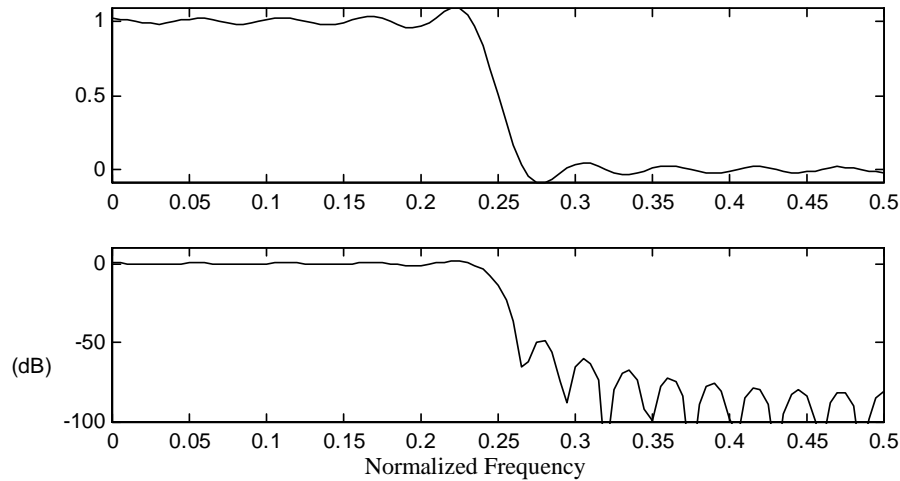
$$10 \log |H(e^{j\omega})|^2 = 10 \log [(1 + \alpha)^2 + 2\alpha \cos \omega] \quad (5.118)$$

as shown in Figure 5.21 for various values of  $\alpha$ .

We see that for  $\alpha > 0$  we have a *low-pass* filter whereas for  $\alpha < 0$  it is a *high-pass* filter, also called a *pre-emphasis* filter, since it emphasizes the high frequencies. In general, filters that boost the high frequencies and attenuate the low frequencies are called *high-pass* filters, and filters that emphasize the low frequencies and de-emphasize the high frequencies are called *low-pass* filters. The parameter  $\alpha$  controls the slope of the curve.

### 5.4.3.3. Window Design FIR Lowpass Filters

The ideal lowpass filter lets all frequencies below  $\omega_0$  go through and eliminates all energy from frequencies above that range. As we described in Section 5.4.1, the ideal lowpass filter has an infinite impulse response, which poses difficulties for implementation in a practical system, as it requires an infinite number of multiplies and adds.

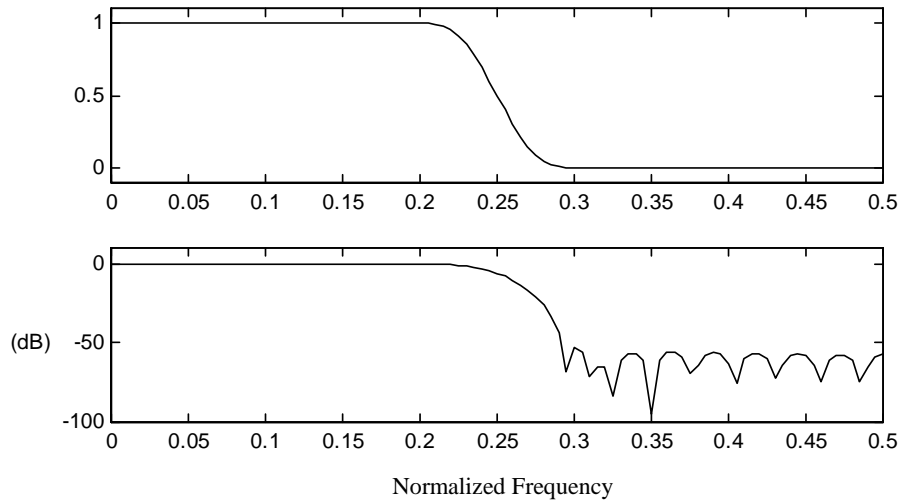


**Figure 5.22** Magnitude frequency response of the truncated sinc signal ( $N=200$ ) for  $\omega_0 = \pi/4$ . It is an approximation to the ideal low-pass filter, though we see that overshoots are present near the transition. The first graph is linear magnitude and the second is in dB.

Since we know that the sinc function decays over time, it is reasonable to assume that a truncated sinc function that keeps a large enough number of samples  $N$  could be a good approximation to the ideal low-pass filter. Figure 5.22 shows the magnitude of the frequency response of such a truncated sinc function for different values of  $N$ . While the approximation gets better for larger  $N$ , the overshoot near  $\omega_0$  doesn't go away and it stays at about

9% of the discontinuity even for large  $N$ . This is known as the *Gibbs phenomenon*, since Yale professor Josiah Gibbs first noticed it in 1899.

In computing the truncated sinc function, we have implicitly multiplied the ideal low-pass filter, the sinc function, by a rectangular window. In the so-called *window design* filter design method, the filter coefficients are obtained by multiplying the ideal sinc function by a tapering window function, such as the Hamming window. The resulting frequency response is the convolution of the ideal lowpass filter function with the transform of the window (shown in Figure 5.23), and it does not exhibit the overshoots seen above, at the expense of a slower transition.



**Figure 5.23** Magnitude frequency response of a low-pass filter obtained with the window design method and a Hamming window ( $N = 200$ ). The first graph is linear magnitude and the second is in dB.

#### 5.4.3.4. Parks McClellan Algorithm

While the window design method is simple, it is hard to predict what the final response will be. Other methods have been proposed whose coefficients are obtained to satisfy some constraints. If our constraints are a maximum ripple of  $\delta_p$  in the *passband* ( $0 \leq \omega < \omega_p$ ), and a minimum attenuation of  $\delta_s$  in the *stopband* ( $\omega_s \leq \omega < \pi$ ), the optimal solution is given by the Parks McClellan algorithm [14].

The transformation

$$x = \cos \omega \quad (5.119)$$

maps the interval  $0 \leq \omega \leq \pi$  into  $-1 \leq x \leq 1$ . We note that

$$\cos(n\omega) = T_n(\cos \omega) \quad (5.120)$$

where  $T_n(x)$  is the  $n^{\text{th}}$ -order *Chebyshev* polynomial. The first two Chebyshev polynomials are given by  $T_0(x) = 1$  and  $T_1(x) = x$ . If we add the following trigonometric identities

$$\begin{aligned} \cos(n+1)\omega &= \cos n\omega \cos \omega - \sin n\omega \sin \omega \\ \cos(n-1)\omega &= \cos n\omega \cos \omega + \sin n\omega \sin \omega \end{aligned} \quad (5.121)$$

and use Eqs. (5.119) and (5.120), we obtain the following recursion formula:

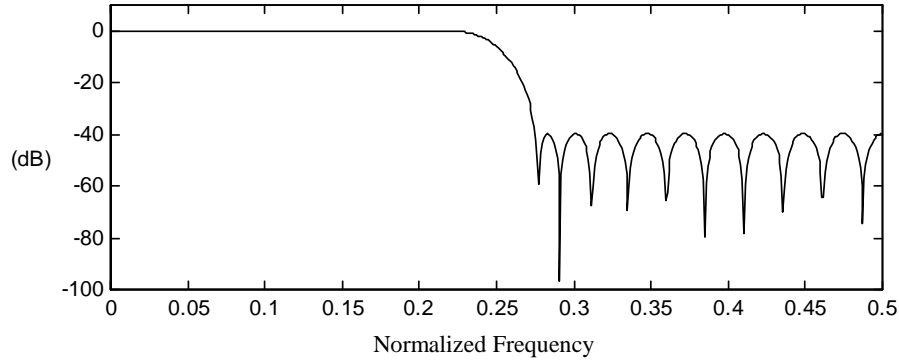
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad \text{for } n > 1 \quad (5.122)$$

Using Eq. (5.120), the magnitude response of a linear phase Type-I filter in Eq. (5.112) can be expressed as an  $L^{\text{th}}$ -order polynomial in  $\cos \omega$ :

$$A(\omega) = \sum_{k=0}^L a_k (\cos \omega)^k \quad (5.123)$$

which, using Eq. (5.119) results in a polynomial

$$P(x) = \sum_{k=0}^L a_k x^k \quad (5.124)$$



**Figure 5.24** Magnitude frequency response of a length-19 lowpass filter designed with the Parks McClellan algorithm.

Given that a desired response is  $D(x) = D(\cos \omega)$ , we define the weighted squared error as

$$E(x) = E(\cos \omega) = W(\cos \omega)[D(\cos \omega) - P(\cos \omega)] = W(x)[D(x) - P(x)] \quad (5.125)$$

where  $W(\cos \omega)$  is the weighting in  $\omega$ . A necessary and sufficient condition for this weighted squared error to be minimized is to have  $P(x)$  alternate between minima and maxima. For the case of a low-pass filter,

$$D(\cos \omega) = \begin{cases} 1 & \cos \omega_p \leq \cos \omega \leq 1 \\ 0 & -1 \leq \cos \omega \leq \cos \omega_s \end{cases} \quad (5.126)$$

and the weight in the stopband is several times larger than in the passband.

These constraints and the response of a filter designed with such a method are shown in Figure 5.24. We can thus obtain a similar transfer function with fewer coefficients using this method.

#### 5.4.4. IIR Filters

Other useful filters are a function of past values of the input and also the output

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{r=0}^M b_r x[n-r] \quad (5.127)$$

whose  $z$ -transform is given by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^M b_r z^{-r}}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (5.128)$$

which in turn can be expressed as a function of the roots of the numerator  $c_r$  (called *zeros*), and denominator  $d_k$  (called *poles*) as

$$H(z) = \frac{Az^{-L} \prod_{r=1}^{M-L} (1 - c_r z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad (5.129)$$

It is not obvious what the impulse response of such a system is by looking at either Eq. (5.128) or Eq. (5.129). To do that, we can compute the inverse  $z$ -transform of Eq. (5.129). If  $M < N$  in Eq. (5.129),  $H(z)$  can be expanded into partial fractions (see Section 5.2.3.3) as

$$H(z) = \sum_{k=1}^N \frac{A_k}{1 - d_k z^{-1}} \quad (5.130)$$

and if  $M \geq N$

$$H(z) = \sum_{k=1}^N \frac{A_k}{1 - d_k z^{-1}} + \sum_{k=0}^{M-N} B_k z^{-k} \quad (5.131)$$

which we can now compute, since we know that the inverse  $z$ -transform of  $H_k(z) = A_k/(1-d_k z^{-1})$  is

$$h_k[n] = \begin{cases} A_k d_k^n u[n] & |d_k| < 1 \\ -A_k d_k^n u[-n-1] & |d_k| > 1 \end{cases} \quad (5.132)$$

so that the convergence region includes the unit circle and therefore  $h_k[n]$  is stable. Therefore, a necessary and sufficient condition for  $H(z)$  to be stable *and* causal simultaneously is that all its poles be inside the unit circle: *i.e.*,  $|d_k| < 1$  for all  $k$ , so that its impulse response is given by

$$h[n] = B_n + \sum_{k=1}^N A_k d_k^n u[n] \quad (5.133)$$

which has an infinite impulse response, and hence its name.

Since IIR systems may have poles outside the unit circle, they are not guaranteed to be stable and causal like their FIR counterparts. This makes IIR filter design more difficult, since only stable and causal filters can be implemented in practice. Moreover, unlike FIR filters, IIR filters do not have linear phase. Despite these difficulties, IIR filters are popular because they are more efficient than FIR filters in realizing steeper roll-offs with fewer coefficients. In addition, as shown in Chapter 6, they represent many physical systems.

#### 5.4.4.1. First-Order IIR Filters

An important type of IIR filter is the first-order filter of the form

$$y[n] = Ax[n] + \alpha y[n-1] \quad (5.134)$$

for  $\alpha$  real. Its transfer function is given by

$$H(z) = \frac{A}{1 - \alpha z^{-1}} \quad (5.135)$$

This system has one pole and no zeros. As we saw in our discussion of  $z$ -transforms in Section 5.2.3, a necessary condition for this system to be both stable and causal is that  $|\alpha| < 1$ . Since for the low-pass filter case  $0 < \alpha < 1$ , it is convenient to define  $\alpha = e^{-b}$  where  $b > 0$ . In addition, the corresponding impulse response is infinite:

$$h[n] = \alpha^n u[n] \quad (5.136)$$

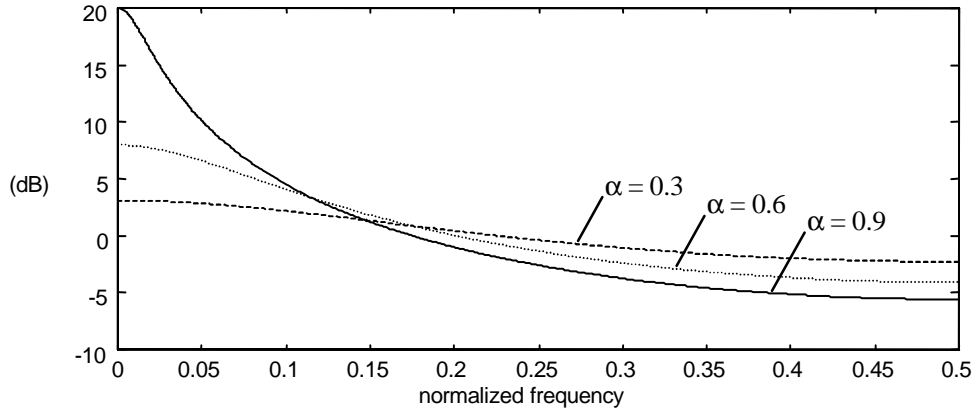
whose Fourier transform is

$$H(e^{j\omega}) = \frac{A}{1 - \alpha e^{-j\omega}} = \frac{A}{1 - e^{-b-j\omega}} \quad (5.137)$$

and magnitude square is given by

$$|H(e^{j\omega})|^2 = \frac{|A|^2}{1 + \alpha^2 - 2\alpha \cos \omega} \quad (5.138)$$

which is shown in Figure 5.25 for  $\alpha > 0$ , which corresponds to a low-pass filter.



**Figure 5.25** Magnitude frequency response of the first-order IIR filter.

The bandwidth of a low-pass filter is defined as the point where its magnitude square is half of its maximum value. Using the first-order Taylor approximation of the exponential function, the following approximation can be used when  $b \rightarrow 0$ :

$$|H(e^{j0})|^2 = \frac{A^2}{|1 - e^{-b}|^2} \approx \frac{A^2}{b^2} \quad (5.139)$$

If the bandwidth  $\omega_b$  is also small, we can similarly approximate

$$|H(e^{j\omega_b})|^2 = \frac{A^2}{|1 - e^{-b-j\omega_b}|^2} \approx \frac{A^2}{|b + j\omega_b|^2} = \frac{A^2}{(b^2 + \omega_b^2)} \quad (5.140)$$

so that for  $\omega_b = b$  we have  $|H(e^{jb})|^2 \approx 0.5 |H(e^{j0})|^2$ . In other words, the bandwidth of this filter equals  $b$ , for small values of  $b$ . The relative error in this approximation<sup>9</sup> is smaller than 2% for  $b < 0.5$ , which corresponds to  $0.6 < \alpha < 1$ . The relationship with the unnormalized bandwidth  $B$  is

$$\alpha = e^{-2\pi B / F_s} \quad (5.141)$$

<sup>9</sup> The exact value is  $\omega_b = \arccos[2 - \cosh b]$ , where  $\cosh b = (e^b + e^{-b})/2$  is the hyperbolic cosine.



For  $\alpha < 0$  it behaves as a high-pass filter, and a similar discussion can be carried out.

#### 5.4.4.2. Second-Order IIR Filters

An important type of IIR filters is the set of second-order filters of the form

$$y[n] = Ax[n] + a_1 y[n-1] + a_2 y[n-2] \quad (5.142)$$

whose transfer function is given by

$$H(z) = \frac{A}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (5.143)$$

This system has two poles and no zeros. A special case is when the coefficients  $A$ ,  $a_1$  and  $a_2$  are real. In this case the two poles are given by

$$z = \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} \quad (5.144)$$

which for the case of  $a_1^2 + 4a_2 > 0$  yields two real roots, and is a degenerate case of two first-order systems. The more interesting case is when  $a_1^2 + 4a_2 < 0$ . In this case we see that the two roots are complex conjugates of each other, which can be expressed in their magnitude and phase notation as

$$z = e^{-\sigma \pm j\omega_0} \quad (5.145)$$

As we mentioned before,  $\sigma > 0$  is a necessary and sufficient condition for the poles to be inside the unit circle and thus for the system to be stable. With those values, the  $z$ -transform is given by

$$H(z) = \frac{A}{(1 - e^{-\sigma + j\omega_0} z^{-1})(1 - e^{-\sigma - j\omega_0} z^{-1})} = \frac{A}{1 - 2e^{-\sigma} \cos(\omega_0) z^{-1} + e^{-2\sigma} z^{-2}} \quad (5.146)$$

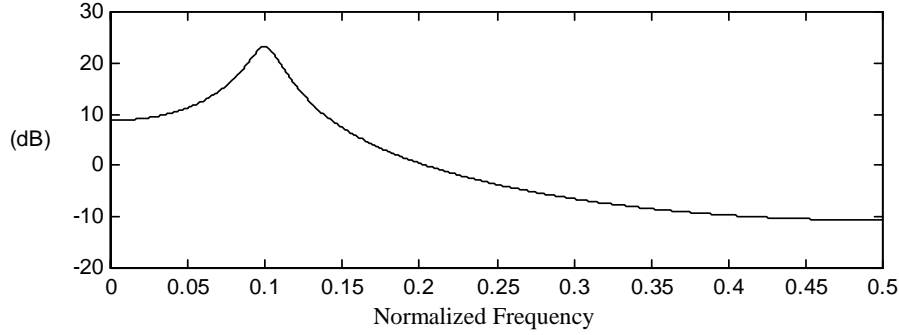
In Figure 5.26 we show the magnitude of its Fourier transform for a value of  $\sigma$  and  $\omega_0$ . We see that the response is centered around  $\omega_0$  and is more concentrated for smaller values of  $\sigma$ . This is a type of *bandpass* filter, since it favors frequencies in a band around  $\omega_0$ . It is left to the reader as an exercise to show that the bandwidth<sup>10</sup> is approximately  $2\sigma$ . The smaller the ratio  $\sigma/\omega_0$ , the sharper the resonance. The filter coefficients can be expressed as a function of the unnormalized bandwidth  $B$  and resonant frequency  $F$  and the sampling frequency  $F_s$  (all expressed in Hz) as

$$a_1 = 2e^{-\pi B/F_s} \cos(2\pi F/F_s) \quad (5.147)$$

<sup>10</sup> The bandwidth of a bandpass filter is the region between half maximum magnitude squared values.

$$a_2 = -e^{-2\pi B / F_s} \quad (5.148)$$

These types of systems are also known as second-order resonators and will be of great use for speech synthesis (Chapter 16), particularly for formant synthesis.



**Figure 5.26** Frequency response of the second-order IIR filter for center frequency of  $F = 0.1F_s$  and bandwidth  $B = 0.01F_s$ .

## 5.5. DIGITAL PROCESSING OF ANALOG SIGNALS

To use the digital signal processing methods, it is necessary to convert the speech signal  $x(t)$ , which is analog, to a digital signal  $x[n]$ , which is formed by periodically sampling the analog signal  $x(t)$  at intervals equally spaced  $T$  seconds apart:

$$x[n] = x(nT) \quad (5.149)$$

where  $T$  is defined as the sampling period, and its inverse  $F_s = 1/T$  as the sampling frequency. In the speech applications considered in this book,  $F_s$  can range from 8000 Hz for telephone applications to 44,100 Hz for high-fidelity audio applications. This section explains the sampling theorem, which essentially says that the analog signal  $x(t)$  can be uniquely recovered given its digital signal  $x[n]$  if the analog signal  $x(t)$  has no energy for frequencies above the *Nyquist* frequency  $F_s/2$ .

We not only prove the sampling theorem, but also provide great insight into the analog-digital conversion, which is used in Chapter 7.

### 5.5.1. Fourier Transform of Analog Signals

The Fourier transform of an analog signal  $x(t)$  is defined as

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt \quad (5.150)$$

with its inverse transform being

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega)e^{j\Omega t} d\Omega \quad (5.151)$$

They are transform pairs. You can prove similar relations for the Fourier transform of analog signals as for their digital signals counterpart.

### 5.5.2. The Sampling Theorem

Let's define  $x_p(t)$

$$x_p(t) = x(t)p(t) \quad (5.152)$$

as a sampled version of  $x(t)$ , where

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT) \quad (5.153)$$

where  $\delta(t)$  is the Dirac delta defined in Section 5.3.2.1. Therefore,  $x_p(t)$  can also be expressed as

$$x_p(t) = \sum_{n=-\infty}^{\infty} x(t)\delta(t-nT) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) = \sum_{n=-\infty}^{\infty} x[n]\delta(t-nT) \quad (5.154)$$

after using Eq. (5.149). In other words,  $x_p(t)$  can be uniquely specified given the digital signal  $x[n]$ .

Using the modulation property of Fourier transforms of analog signals, we obtain

$$X_p(\Omega) = \frac{1}{2\pi} X(\Omega) * P(\Omega) \quad (5.155)$$

Following a derivation similar to that in Section 5.3.2.2, one can show that the transform of the impulse train  $p(t)$  is given by

$$P(\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) \quad (5.156)$$

where  $\Omega_s = 2\pi F_s$  and  $F_s = 1/T$ , so that

$$X_p(\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(\Omega - k\Omega_s) \quad (5.157)$$

From Figure 5.27 it can be seen that if

$$X(\Omega) = 0 \text{ for } |\Omega| > \Omega_s / 2 \quad (5.158)$$

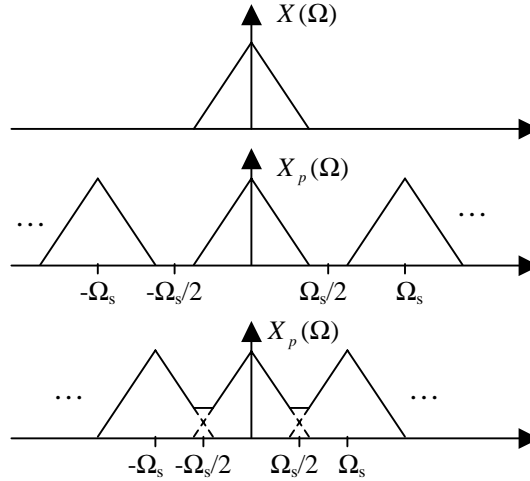
then  $X(\Omega)$  can be completely recovered from  $X_p(\Omega)$  as follows

$$X(\Omega) = R_{\Omega_s}(\Omega) X_p(\Omega) \quad (5.159)$$

where

$$R_{\Omega_s}(\Omega) = \begin{cases} 1 & |\Omega| < \Omega_s / 2 \\ 0 & \text{otherwise} \end{cases} \quad (5.160)$$

is an ideal lowpass filter. We can also see that if Eq. (5.158) is not met, then *aliasing* will take place and  $X(\Omega)$  can no longer be recovered from  $X_p(\Omega)$ . Since, in general, we cannot be certain that Eq. (5.158) is true, the analog signal is low-pass filtered with an ideal filter given by Eq. (5.160), which is called anti-aliasing filter, prior to sampling. Limiting the bandwidth of our analog signal is the price we have to pay to be able to manipulate it digitally.



**Figure 5.27**  $X(\Omega)$ ,  $X_p(\Omega)$  for the case of no aliasing and aliasing.

The inverse Fourier transform of Eq. (5.160), computed through Eq. (5.151), is a sinc function

$$r_T(t) = \text{sinc}(t/T) = \frac{\sin(\pi t/T)}{\pi t/T} \quad (5.161)$$

so that using the convolution property in Eq. (5.159) we obtain

$$x(t) = r_T(t) * x_p(t) = r_T(t) * \sum_{k=-\infty}^{\infty} x[k] \delta(t - kT) = \sum_{k=-\infty}^{\infty} x[k] r_T(t - kT) \quad (5.162)$$

The *sampling theorem* states that we can recover the continuous time signal  $x(t)$  just from its samples  $x[n]$  using Eqs. (5.161) and (5.162). The angular frequency  $\Omega_s = 2\pi F_s$  is expressed in terms of the sampling frequency  $F_s$ .  $T = 1/F_s$  is the sampling period, and  $F_s/2$  the *Nyquist* frequency. Equation (5.162) is referred to as *bandlimited interpolation* because  $x(t)$  is reconstructed by interpolating  $x[n]$  with sinc functions that are bandlimited.

Now let's see the relationship between  $X_p(\Omega)$  and  $X(e^{j\omega})$ , the Fourier transform of the discrete sequence  $x[n]$ . From Eq. (5.154) we have

$$X_p(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n T} \quad (5.163)$$

so that the continuous transform  $X_p(\Omega)$  equals the discrete Fourier transform  $X(e^{j\omega})$  at  $\omega = \Omega T$ .

### 5.5.3. Analog-to-Digital Conversion

The process of converting an analog signal  $x(t)$  into a digital signal  $x[n]$  is called *Analog-to-Digital conversion*, or A/D for short, and the device that does it called an *Analog-to-Digital Converter*. In Section 5.5.2 we saw that an ideal low-pass anti-aliasing filter was required on the analog signal, which of course is not realizable in practice so that an approximation has to be used. In practice, sharp analog filters can be implemented on the same chip using switched capacitor filters, which have attenuations above 60 dB in the stop band so that aliasing tends not to be an important issue for speech signals. The passband is not exactly flat, but this again does not have much significance for speech signals (for other signals, such as those used in modems, this issue needs to be studied more carefully).

Although such sharp analog filters are possible, they can be expensive and difficult to implement. One common solution involves the use of a simple analog low-pass filter with a large attenuation at  $MF_s/2$ , a multiple of the required cutoff frequency. Then *over-sampling* is done at the new rate  $MF_s$ , followed by a sharper digital filter with a cut-off frequency of  $F_s/2$  and downsampling (see Section 5.6). This is equivalent to having used a sharp analog filter, with the advantage of a lower-cost implementation. This method also allows variable sampling rates with minimal increase in cost and complexity. This topic is discussed in more detail in Chapter 7 in the context of sigma-delta modulators.

In addition, the pulses in Eq. (5.59) cannot be zero length in practice, and therefore the sampling theorem does not hold. However, current hardware allows the pulses to be small enough that the analog signal can be approximately recovered. The signal level is then maintained during  $T$  seconds, while the conversion to digital is being carried out.

A real A/D converter cannot provide real numbers for  $x[n]$ , but rather a set of integers typically represented with 16 bits, which gives a range between  $-32,768$  and  $32,767$ . Such conversion is achieved by comparing the analog signal to a number of different signal levels. This means that *quantization noise* has been added to the digital signal. This is typically not a big problem for speech signals if using 16 bits or more since, as is shown in Chapter 7, other noises will mask the quantization noise anyway. Typically, quantization noise becomes an issue only if 12 or fewer bits are used. A more detailed study of the effects of quantization is presented in Chapter 7.

Finally, A/D subsystems are not exactly linear, which adds another source of distortion. This nonlinearity can be caused by, among things, jitter and drift in the pulses and unevenly spaced comparators. For popular A/D subsystems, such as *sigma-delta* A/D, an offset is typically added to  $x[n]$ , which in practice is not very important, because speech signals do not contain information at  $f = 0$ , and thus can be safely ignored.

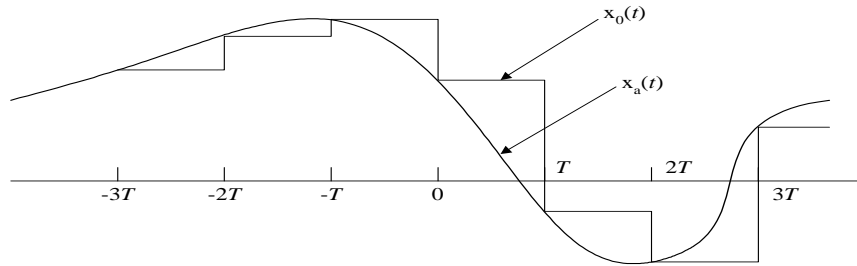
#### 5.5.4. Digital-to-Analog Conversion

The process of converting the digital signal  $x[n]$  back into an analog  $x(t)$  is called *digital-to-analog conversion*, or D/A for short. The ideal band-limited interpolation requires ideal sinc functions as shown in Eq. (5.162), which are not realizable. To convert the digital signal to analog, a zero-order hold filter

$$h_0(t) = \begin{cases} 1 & 0 < t < T \\ 0 & \text{otherwise} \end{cases} \quad (5.164)$$

is often used, which produces an analog signal as shown in Figure 5.28. The output of such a filter is given by

$$x_0(t) = h_0(t) * \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT) = \sum_{n=-\infty}^{\infty} x[n]h_0(t - nT) \quad (5.165)$$



**Figure 5.28** Output of a zero-order hold filter.

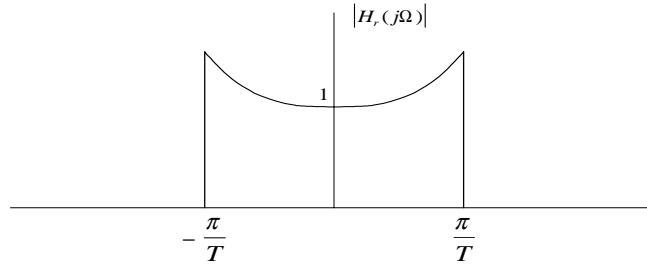
The Fourier transform of the zero-hold filter in Eq. (5.164) is, using Eq. (5.150),

$$H_0(\Omega) = \frac{2 \sin(\Omega T / 2)}{\Omega} e^{-j\Omega T / 2} \quad (5.166)$$

and, since we need an ideal lowpass filter to achieve the band-limited interpolation of Eq. (5.162), the signal  $x_0(t)$  has to be filtered with a reconstruction filter with transfer function

$$H_r(\Omega) = \begin{cases} \frac{\Omega T / 2}{\sin(\Omega T / 2)} e^{j\Omega T / 2} & |\Omega| < \pi / T \\ 0 & |\Omega| > \pi / T \end{cases} \quad (5.167)$$

In practice, the phase compensation is ignored, as it amounts to a delay of  $T/2$  seconds. Its magnitude response can be seen in Figure 5.29. In practice, such an analog filter is not realizable and an approximation is made. Since the zero-order hold filter is already low-pass, the reconstruction filter doesn't need to be that sharp.



**Figure 5.29** Magnitude frequency response of the reconstruction filter used in digital-to-analog converters after a zero-hold filter.

In the above discussion we note that practical A/D and D/A systems introduce distortions, which causes us to wonder whether it is a good idea to go through this process just to manipulate digital signals. It turns out that for most speech processing algorithms described in Chapter 6, the advantages of operating with digital signals outweigh the disadvantage of the distortions described above. Moreover, commercial A/D and D/A systems are such that the errors and distortions can be arbitrarily small. The fact that music in digital format (as in compact discs) has won out over analog format (cassettes) shows that this is indeed the case. Nonetheless, it is important to be aware of the above limitations when designing a system.

## 5.6. MULTIRATE SIGNAL PROCESSING

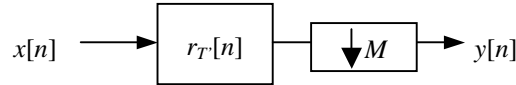
The term *Multirate Signal Processing* refers to processing of signals sampled at different rates. A particularly important problem is that of sampling-rate conversion. It is often the case that we have a digital signal  $x[n]$  sampled at a sampling rate  $F_s$ , and we want to obtain an equivalent signal  $y[n]$  but at a different sampling rate  $F'_s$ . This often occurs in A/D systems that oversample in order to use smaller quantizers, such as a delta or sigma delta-

quantizer (see Chapter 7), and a simpler analog filter, and then have to downsample the signal. Other examples include mixing signals of different sampling rates and downsampling to reduce computation (many signal processing algorithms have a computational complexity proportional to the sampling rate or its square).

A simple solution is to convert the digital signal  $x[n]$  into an analog signal  $x(t)$  with a D/A system running at  $F_s$  and then convert it back to digital with an A/D system running at  $F'_s$ . An interesting problem is whether this could be done in the digital domain directly, and the techniques to do so belong to the general class of multi-rate processing.

### 5.6.1. Decimation

If we want to reduce the sampling rate by a factor of  $M$ , i.e.,  $T' = MT$ , we take every  $M$  samples. In order to avoid aliasing, we need to lowpass filter the signal to bandlimit it to frequencies  $1/T'$ . This is shown in Figure 5.30, where the arrow pointing down indicates the decimation.



**Figure 5.30** Block diagram of the decimation process.

Since the output is not desired at all instants  $n$ , but only every  $M$  samples, the computation can be reduced by a factor of  $M$  over the case where lowpass filtering is done first and *decimation* later. To do this we express the analog signal  $x_i(t)$  at the output of the lowpass filter as

$$x_i(t) = \sum_{k=-\infty}^{\infty} x[k]r_{T'}(t - kT) \quad (5.168)$$

and then look at the value  $t' = nT'$ . The decimated signal  $y[n]$  is then given by

$$y[n] = x_i(nT') = \sum_{k=-\infty}^{\infty} x[k]r_{T'}(nT' - kT) = \sum_{k=-\infty}^{\infty} x[k]\text{sinc}\left(\frac{Mn - k}{M}\right) \quad (5.169)$$

which can be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[Mn - k] \quad (5.170)$$

where

$$h[n] = \text{sinc}(n/M) \quad (5.171)$$



In practice, the ideal lowpass filter  $h[n]$  is approximated by an FIR filter with a cutoff frequency of  $1/(2M)$ .

### 5.6.2. Interpolation

If we want to increase the sampling rate by a factor of  $N$ , so that  $T' = T/N$ , we do not have any aliasing and no further filtering is necessary. In fact we already know one out of every  $N$  output samples

$$y[Nn] = x[n] \quad (5.172)$$

and we just need to compute the  $(N-1)$  samples in-between. Since we know that  $x[n]$  is a bandlimited signal, we can use the sampling theorem in Eq. (5.162) to reconstruct the analog signal as

$$x_t(t) = \sum_{k=-\infty}^{\infty} x[k]r_T(t-kT) \quad (5.173)$$

and thus the interpolated signal  $y[n]$  as

$$y[n] = x(nT') = \sum_{k=-\infty}^{\infty} x[k]r_T(nT' - kT) = \sum_{k=-\infty}^{\infty} x[k]\text{sinc}\left(\frac{n-kN}{N}\right) \quad (5.174)$$

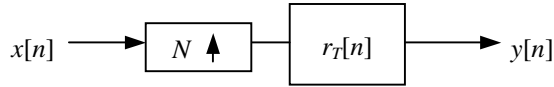
Now let's define

$$x'[k'] = \begin{cases} x[Nk] & k' = Nk \\ 0 & \text{otherwise} \end{cases} \quad (5.175)$$

which, inserted into Eq. (5.174), gives

$$y[n] = \sum_{k'=-\infty}^{\infty} x'[k']\text{sinc}((n-k')/N) \quad (5.176)$$

This can be seen in Figure 5.31, where the block with the arrow pointing up implements Eq. (5.175).



**Figure 5.31** Block diagram of the interpolation process.

Equation (5.174) can be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-kN] \quad (5.177)$$

where we have defined

$$h[n] = \text{sinc}(n/N) \quad (5.178)$$

Again, in practice, the ideal low-pass filter  $h[n]$  is approximated by an FIR filter with a cutoff frequency of  $1/(2N)$ .

### 5.6.3. Resampling

To resample the signal so that  $T' = TM/N$ , or  $F'_s = F_s(N/M)$ , we can first upsample the signal by  $N$  and then downsample it by  $M$ . However, there is a more efficient way. Proceeding similarly to decimation and interpolation, one can show the output is given by

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[nM - kN] \quad (5.179)$$

where

$$h[n] = \text{sinc}\left(\frac{n}{\max(N, M)}\right) \quad (5.180)$$

for the ideal case. In practice,  $h[n]$  is an FIR filter with a cutoff frequency of  $1/(2\max(N, M))$ . We can see that Eq. (5.179) is a superset of Eqs. (5.170) and (5.177).

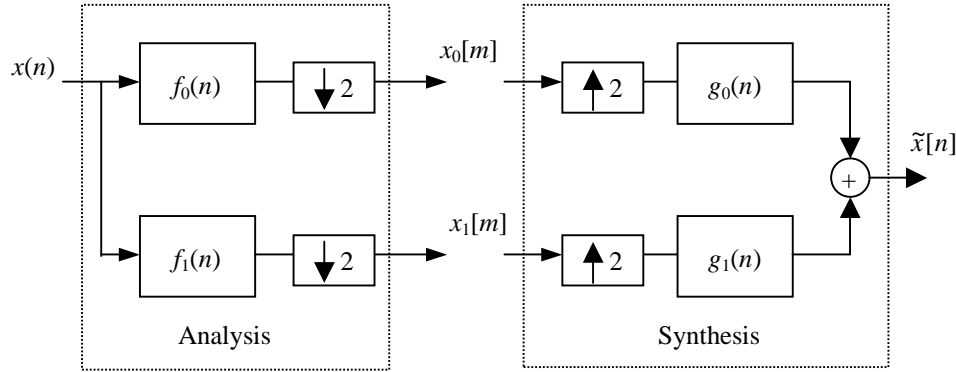
## 5.7. FILTERBANKS

A filterbank is a collection of filters that span the whole frequency spectrum. In this section we describe the fundamentals of filterbanks, which are used in speech and audio coding, echo cancellation, and other applications. We first start with a filterbank with two equal bands, then explain multi-resolution filterbanks, and present the FFT as a filterbank. Finally we introduce the concept of lapped transforms and wavelets.

### 5.7.1. Two-Band Conjugate Quadrature Filters

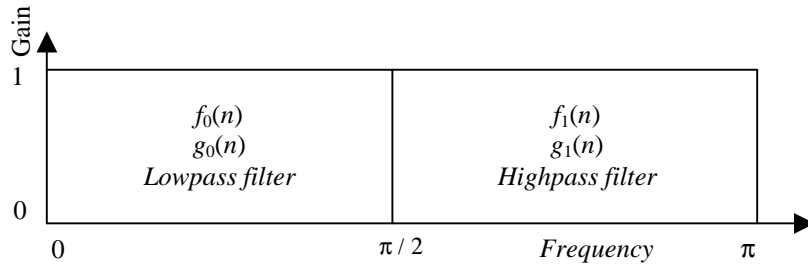
A two-band filterbank is shown in Figure 5.32, where the filters  $f_0[n]$  and  $g_0[n]$  are low-pass filters, and the filters  $f_1[n]$  and  $g_1[n]$  are high-pass filters, as shown in Figure 5.33. Since the output of  $f_0[n]$  has a bandwidth half of that of  $x[n]$ , we can sample it at half the rate of  $x[n]$ . We do that by decimation (throwing out every other sample), as shown in Figure 5.32. The output of such a filter plus decimation is  $x_0[m]$ . Similar results can be shown for  $f_1[n]$  and  $x_1[n]$ .

For reconstruction, we upsample  $x_0[m]$ , by inserting a 0 between every sample. Then we low-pass filter it with filter  $g_0[n]$  to complete the interpolation, as we saw in Section 5.6. A similar process can be done with the high pass filters  $f_1[n]$  and  $g_1[n]$ . Adding the two bands produces  $\tilde{x}[n]$ , which is identical to  $x[n]$  if the filters are ideal.



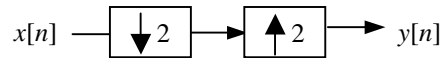
**Figure 5.32** Two-band filterbank.

In practice, however, ideal filters such as those in Figure 5.33 are not achievable, so we would like to know if it is possible to build a filterbank that has perfect reconstruction with FIR filters. The answer is affirmative, and in this section we describe conjugate quadrature filters, which are the basis for the solutions.



**Figure 5.33** Ideal frequency responses of analysis and synthesis filters for the two-band filterbank.

To investigate this, let's analyze the cascade of a downsampler and an upsampler (Figure 5.34). The output  $y[n]$  is a signal whose odd samples are zero and whose even samples are the same as those of the input signal  $x[n]$ .



**Figure 5.34** Cascade of a downsampler and an upsampler.

The  $z$ -transform of the output is given by

$$\begin{aligned} Y(z) &= \sum_{\substack{n=-\infty \\ n \text{ even}}}^{\infty} x[n]z^{-n} = \frac{1}{2} \sum_{n=-\infty}^{\infty} x[n]z^{-n} + \frac{1}{2} \sum_{n=-\infty}^{\infty} (-1)^n x[n]z^{-n} \\ &= \frac{X(z) + X(-z)}{2} \end{aligned} \quad (5.181)$$

Using Eq. (5.181) and the system in Figure 5.32, we can express the  $z$ -transform of the output in Figure 5.32 as

$$\begin{aligned} \tilde{X}(z) &= \left( \frac{F_0(z)G_0(z) + F_1(z)G_1(z)}{2} \right) X(z) \\ &+ \left( \frac{F_0(-z)G_0(z) + F_1(-z)G_1(z)}{2} \right) X(-z) \\ &= \left( \frac{F_0(z)X(z) + F_0(-z)X(-z)}{2} \right) G_0(z) + \left( \frac{F_1(z)X(z) + F_1(-z)X(-z)}{2} \right) G_1(z) \end{aligned} \quad (5.182)$$

which for perfect reconstruction requires the output to be a delayed version of the input, and thus

$$\begin{aligned} F_0(z)G_0(z) + F_1(z)G_1(z) &= 2z^{-(L-1)} \\ F_0(-z)G_0(z) + F_1(-z)G_1(z) &= 0 \end{aligned} \quad (5.183)$$

These conditions are met if we select the so-called *Conjugate Quadrature Filters* (CQF) [17], which are FIR filters that specify  $f_1[n]$ ,  $g_0[n]$ , and  $g_1[n]$  as a function of  $f_0[n]$ :

$$\begin{aligned} f_1[n] &= (-1)^n f_0[L-1-n] \\ g_0[n] &= f_0[L-1-n] \\ g_1[n] &= f_1[L-1-n] \end{aligned} \quad (5.184)$$

where  $f_0[n]$  is an FIR filter of even length  $L$ . The  $z$ -transforms of Eq. (5.184) are

$$\begin{aligned} F_1(z) &= z^{-(L-1)} F_0(-z^{-1}) \\ G_0(z) &= z^{-(L-1)} F_0(z^{-1}) \\ G_1(z) &= F_0(-z) \end{aligned} \quad (5.185)$$

so that the second equation in Eq. (5.183) is met if  $L$  is even. In order to analyze the first equation in Eq. (5.183), let's define  $P(z)$  as

$$\begin{aligned}
P(z) &= F_0(z)F_0(z^{-1}) \\
p[n] &= \sum_m f_0[m]f_0[m+n]
\end{aligned} \tag{5.186}$$

then insert Eq. (5.185) into (5.183), use Eq. (5.186), and obtain the following condition:

$$P(z) + P(-z) = 2 \tag{5.187}$$

Taking the inverse  $z$ -transform of Eq. (5.186) and using Eq. (5.181), we obtain

$$p[n] = \begin{cases} 1 & n = 0 \\ 0 & n = 2k \end{cases} \tag{5.188}$$

so that all even samples of the autocorrelation of  $f_0[n]$  are zero, except for  $n = 0$ . Since  $f_0[n]$  is a half-band low-pass filter,  $p[n]$  is also a half-band low-pass filter. The ideal half-band filter  $h[n]$

$$h[n] = \frac{\sin(\pi n / 2)}{\pi n} \tag{5.189}$$

satisfies Eq. (5.188), as does any half-band zero-phase filter (a linear phase filter with no delay). Therefore, the steps to build CQF are

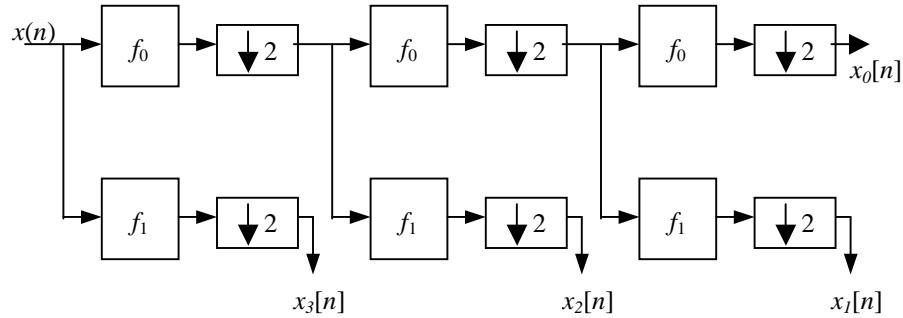
1. Design a  $(2L - 1)$  tap<sup>11</sup> half-band linear-phase low-pass filter  $p[n]$  with any available technique, for an even value of  $L$ . For example, one could use the Parks McClellan algorithm, constraining the passband and stopband cutoff frequencies so that  $\omega_p = \pi - \omega_s$  and using an error weighting that is the same for the passband and stopband. This results in a half-band linear-phase filter with equal ripple  $\delta$  in both bands. Another possibility is to multiply the ideal half-band filter in Eq. (5.189) by a window with low-pass characteristics.
2. Add a value  $\delta$  to  $p[0]$  so that we can guarantee that  $P(e^{j\omega}) \geq 0$  for all  $\omega$  and thus is a legitimate power spectral density.
3. Spectrally factor  $P(z) = F_0(z)F_0(z^{-1})$  by computing its roots.
4. Compute  $f_1[n]$ ,  $g_0[n]$  and  $g_1[n]$  from Eq. (5.184).

### 5.7.2. Multiresolution Filterbanks

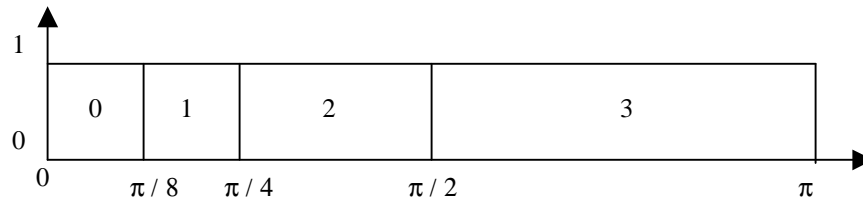
While the above filterbank has equal bandwidth for both filters, it may be desirable to have varying bandwidths, since it has been proven to work better in speech recognition systems. In this section we show how to use the two-band conjugate quadrature filters described in the previous section to design a filterbank with more than two bands. In fact, multi-

<sup>11</sup> A filter with  $N$  taps is a filter of length  $N$ .

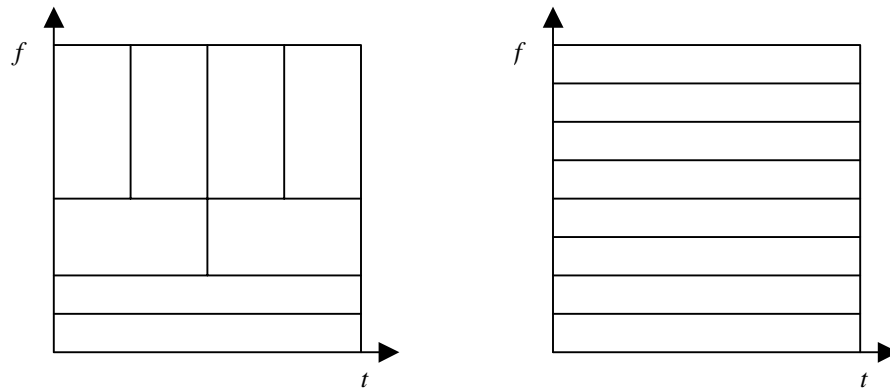
resolution analysis such as that of Figure 5.35, are possible with bands of different bandwidths (see Figure 5.36).



**Figure 5.35** Analysis part of a multi-resolution filterbank designed with conjugate quadrature filters. Only  $f_0[n]$  needs to be specified.



**Figure 5.36** Ideal frequency responses of the multi-resolution filterbank of Figure 5.35. Note that  $x_0[n]$  and  $x_1[n]$  occupy 1/8 of the total bandwidth.



**Figure 5.37** Two different time-frequency tilings: the non-uniform filterbank and that obtain through a short-time Fourier transform. Notice that the area of each tile is constant.

One interesting result is that the product of time resolution and frequency resolution is constant (all the tiles in Figure 5.37 have the same area), since filters with smaller bandwidths do not need to be sampled as often. Instead of using Fourier basis for decomposition, multi-resolution filterbanks allow more flexibility in the tiling of the time-frequency plane.

### 5.7.3. The FFT as a Filterbank

It turns out that we can use the Fourier transform to construct a filterbank. To do that, we decompose the input signal  $x[n]$  as a sum of *short-time* signals  $x_m[n]$

$$x[n] = \sum_{m=-\infty}^{\infty} x_m[n] \quad (5.190)$$

where  $x_m[n]$  is obtained as

$$x_m[n] = x[n]w_m[n] \quad (5.191)$$

the product of  $x[n]$  by a *window* function  $w_m[n]$  of length  $N$ . From Eqs. (5.190) and (5.191) we see that the window function has to satisfy

$$\sum_{m=-\infty}^{\infty} w_m[n] = 1 \quad \forall n \quad (5.192)$$

If the short-term signals  $x_m[n]$  are spaced  $M$  samples apart, we define the window  $w_m[n]$  as:

$$w_m[n] = w[n - Mm] \quad (5.193)$$

where  $w[n] = 0$  for  $n < 0$  and  $n > N$ . The windows  $w_m[n]$  overlap in time while satisfying Eq. (5.192).

Since  $x_m[n]$  has  $N$  nonzero values, we can evaluate its length- $N$  DFT as

$$\begin{aligned} X_m[k] &= \sum_{l=0}^{N-1} x_m[Mm + l] e^{-j\omega_k l} \\ &= \sum_{l=0}^{N-1} x[Mm + l] w[l] e^{-j\omega_k l} = \sum_{l=0}^{N-1} x[Mm + l] f_k[-l] \end{aligned} \quad (5.194)$$

where  $\omega_k = 2\pi k / N$  and the analysis filters  $f_k[l]$  are given by

$$f_k[l] = w[-l] e^{j\omega_k l} \quad (5.195)$$

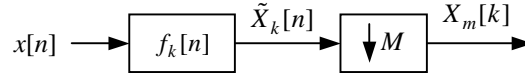
If we define  $\tilde{X}_k[n]$  as

$$\tilde{X}_k[n] = x[n] * f_k[n] = \sum_{r=-\infty}^{\infty} x[n-r]f_k[r] = \sum_{l=0}^{N-1} x[n+l]f_k[-l] \quad (5.196)$$

then Eqs. (5.194) and (5.196) are related by

$$X_m[k] = \tilde{X}_k[mM] \quad (5.197)$$

This manipulation is shown in Figure 5.38, so that the DFT output  $X_m[k]$  is  $\tilde{X}_k[n]$  decimated by  $M$ .



**Figure 5.38** Fourier analysis used to build a linear filter.

The short-time signal  $x_m[n]$  can be recovered through the inverse DFT of  $X_m[k]$  as

$$x_m[mM + l] = h[l] \sum_{k=0}^{N-1} X_m[k] e^{j\omega_k l} \quad (5.198)$$

where  $h[n]$  has been defined as

$$h[n] = \begin{cases} 1/N & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases} \quad (5.199)$$

so that Eq. (5.198) is valid for all values of  $l$ , and not just  $0 \leq l < N$ .

Making the change of variables  $mM + l = n$  in Eq. (5.198) and inserting it into Eq. (5.190) results in

$$\begin{aligned} x[n] &= \sum_{m=-\infty}^{\infty} h[n - mM] \sum_{k=0}^{N-1} X_m[k] e^{j\omega_k (n - mM)} \\ &= \sum_{k=0}^{N-1} \sum_{m=-\infty}^{\infty} X_m[k] g_k[n - mM] \end{aligned} \quad (5.200)$$

where the synthesis filters  $g_k[n]$  are defined as

$$g_k[n] = h[n] e^{j\omega_k n} \quad (5.201)$$

Now, let's define the upsampled version of  $X_m[k]$  as

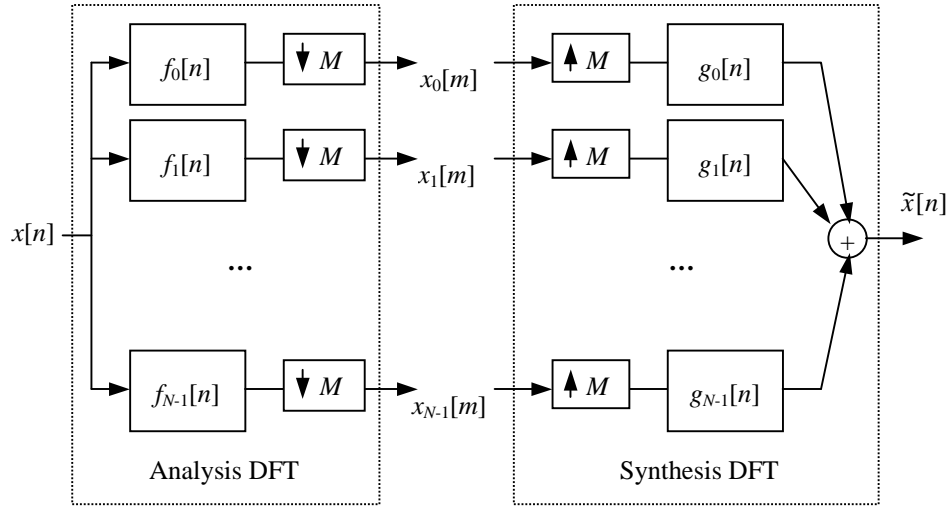
$$\hat{X}_k[l] = \begin{cases} X_m[k] & l = mM \\ 0 & \text{otherwise} \end{cases} \quad (5.202)$$



which, inserted into Eq. (5.200), yields

$$x[n] = \sum_{k=0}^{N-1} \sum_{l=-\infty}^{\infty} \hat{X}_k[l] g_k[n-l] = \sum_{k=0}^{N-1} \hat{X}_k[n] * g_k[n] \quad (5.203)$$

Thus, the signal can be reconstructed. The block diagram of the analysis/resynthesis filterbank implemented by the DFT can be seen in Figure 5.39, where  $x_k[m] = X_m[k]$  and  $\tilde{x}[n] = x[n]$ .



**Figure 5.39** A filterbank with  $N$  analysis and synthesis filters.

For perfect reconstruction we need  $N \geq M$ . If  $w[n]$  is a rectangular window of length  $N$ , the frame rate has to be  $M = N$ . We can also use overlapping windows with  $N = 2M$  (50% overlap), such as Hamming or Hanning windows, and still get perfect reconstruction. The use of such overlapping windows increases the data rate by a factor of 2, but the analysis filters have much less spectral leakage because of the higher attenuation of the Hamming/Hanning window outside the main lobe.

#### 5.7.4. Modulated Lapped Transforms

The filterbank of Figure 5.39 is useful because, as we see in Chapter 7, it is better to quantize the spectral coefficients than the waveform directly. If the DFT coefficients are quantized, there will be some discontinuities at frame boundaries. To solve this problem we can distribute the window  $w[n]$  between the analysis and synthesis filters so that

$$w[n] = w_a[n] w_s[n] \quad (5.204)$$

so that the analysis filters are given by

$$f_k[n] = w_a[-n]e^{j\omega_k n} \quad (5.205)$$

and the synthesis filters by

$$g_k[n] = w_s[n]e^{-j\omega_k n} \quad (5.206)$$

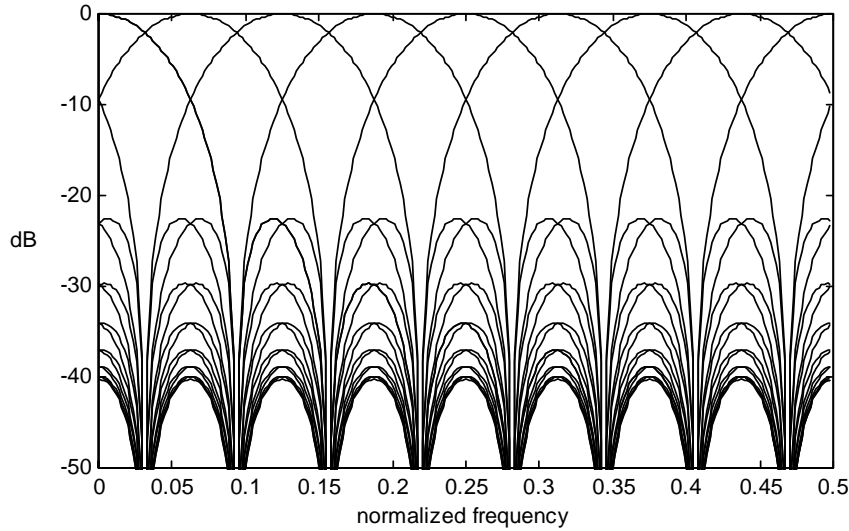
This way, if there is a quantization error, the use of a tapering synthesis window will substantially decrease the border effect. A common choice is  $w_a[n] = w_s[n]$ , which for the case of  $w[n]$  being a Hanning window divided by  $N$ , results in

$$w_a[n] = w_s[n] = \frac{1}{\sqrt{N}} \sin\left(\frac{\pi n}{N}\right) \quad \text{for } 0 \leq n < N \quad (5.207)$$

so that the analysis and synthesis filters are the reversed versions of each other:

$$f_k[-n] = g_k[n] = \frac{\sin(\pi n / N)}{\sqrt{N}} e^{j2\pi nk / N} \Pi_N[n] = h_k^N[n] \quad (5.208)$$

whose frequency response can be seen in Figure 5.40.

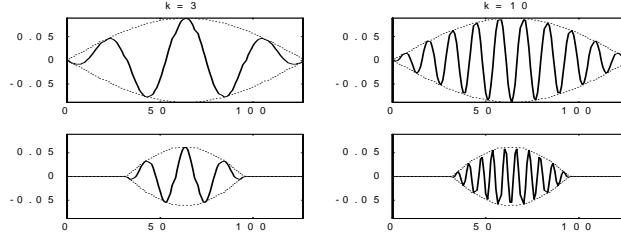


**Figure 5.40** Frequency response of the Lapped Orthogonal Transform filterbank.

The functions  $h_k^N[n]$  in Eq. (5.208) are sine modulated complex exponentials, which have the property

$$h_k^{N/2}[n] = 2^{-1/2} h_k^N[2n] \quad (5.209)$$

which is a property typical of functions called *wavelets*, i.e., they can be obtained from each other by stretching by 2 and scaling them appropriately. Such wavelets can be seen in Figure 5.41.



**Figure 5.41** Iterations of the wavelet  $h_k^N[n]$  for several values of  $k$  and  $N$ .

If instead of modulating a complex exponential we use a cosine sequence, we obtain the *Modulated Lapped Transform* (MLT) [7], also known as the *Modified Discrete Cosine Transform* (MDCT):

$$p_{kn} = f_k[2M-1-n] = g_k[n] = h[n] \sqrt{\frac{2}{M}} \cos \left[ \left( k + \frac{1}{2} \right) \left( n + \frac{M+1}{2} \right) \frac{\pi}{M} \right] \quad (5.210)$$

for  $k = 0, 1, \dots, M-1$  and  $n = 0, 1, \dots, 2M-1$ . There are  $M$  filters with  $2M$  taps each, and  $h[n]$  is a symmetric window  $h[n] = h[2M-1-n]$  that satisfies

$$h^2[n] + h^2[n+M] = 1 \quad (5.211)$$

where the most common choice for  $h[n]$  is

$$h[n] = \sin \left[ \left( n + \frac{1}{2} \right) \frac{\pi}{2M} \right] \quad (5.212)$$

A fast algorithm can be used to compute these filters based on the DCT, which is called the *Lapped Orthogonal Transform* (LOT).

## 5.8. STOCHASTIC PROCESSES

While in this chapter we have been dealing with *deterministic signals*, we also need to deal with *noise*, such as the static present in a poorly tuned AM station. To analyze noise signals we need to introduce the concept of stochastic processes, also known as random processes. A *discrete-time* stochastic process  $\mathbf{x}[n]$ , also denoted by  $\mathbf{x}_n$ , is a sequence of random variables for each time instant  $n$ . *Continuous-time* stochastic processes  $\mathbf{x}(t)$ , random variables for each value of  $t$ , will not be the focus of this book, though their treatment is similar to that of discrete-time processes. We use bold for random variables and regular text for deterministic signals.

Here, we cover the statistics of stochastic processes, defining stationary and ergodic processes and the output of linear systems to such processes.

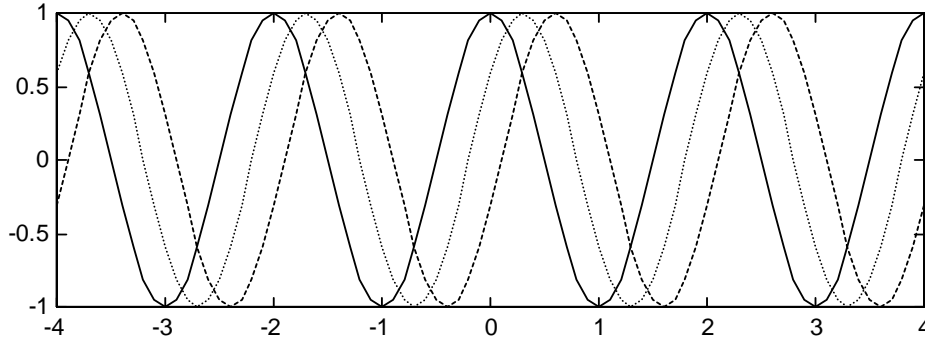
---

**Example**

We can define a random process  $\mathbf{x}[n]$  as

$$\mathbf{x}[n] = \cos[\omega n + \phi] \quad (5.213)$$

where  $\phi$  is real random variable with a uniform pdf in the interval  $(-\pi, \pi)$ . Several realizations of this random process are displayed in Figure 5.42.



**Figure 5.42** Several realizations of a sinusoidal random process with a random phase.

---

### 5.8.1. Statistics of Stochastic Processes

In this section we introduce several statistics of stochastic processes such as distribution, density function, mean and autocorrelation. We also define several types of processes depending on these statistics.

For a specific  $n$ ,  $\mathbf{x}[n]$  is a random variable with *distribution*

$$F(x, n) = P\{\mathbf{x}[n] \leq x\} \quad (5.214)$$

Its first derivative with respect to  $x$  is the first-order density function, or simply the *probability density function* (pdf)

$$f(x, n) = \frac{dF(x, n)}{dx} \quad (5.215)$$

The second-order distribution of the process  $\mathbf{x}[n]$  is the joint distribution

$$F(x_1, x_2; n_1, n_2) = P\{\mathbf{x}[n_1] \leq x_1, \mathbf{x}[n_2] \leq x_2\} \quad (5.216)$$

of the random variables  $\mathbf{x}[n_1]$  and  $\mathbf{x}[n_2]$ . The corresponding density equals

$$f(x_1, x_2; n_1, n_2) = \frac{\partial^2 F(x_1, x_2; n_1, n_2)}{\partial x_1 \partial x_2} \quad (5.217)$$

A complex random process  $\mathbf{x}[n] = \mathbf{x}_r[n] + j\mathbf{x}_i[n]$  is specified in terms of the joint statistics of the real processes  $\mathbf{x}_r[n]$  and  $\mathbf{x}_i[n]$ .

The *mean*  $\mu[n]$  of  $\mathbf{x}[n]$ , also called first-order moment, is defined as the expected value of the random variable  $\mathbf{x}[n]$  for each value of  $n$ :

$$\mu_x[n] = E\{\mathbf{x}[n]\} = \int_{-\infty}^{\infty} \mathbf{x}[n] f(\mathbf{x}, n) d\mathbf{x} \quad (5.218)$$

The autocorrelation of complex random process  $\mathbf{x}[n]$ , also called second-order moment, is defined as

$$R_{xx}[n_1, n_2] = E\{\mathbf{x}[n_1] \mathbf{x}^*[n_2]\} = R_{xx}^*[n_2, n_1] \quad (5.219)$$

which is a statistical average, unlike the autocorrelation of a deterministic signal defined in Eq. (5.45), which was an average over time.

---

### Example

Let's look at the following sinusoidal random process

$$\mathbf{x}[n] = \mathbf{r} \cos[\omega n + \phi] \quad (5.220)$$

where  $\mathbf{r}$  and  $\phi$  are independent and  $\phi$  is uniform in the interval  $(-\pi, \pi)$ . This process is *zero-mean* because

$$\mu_x[n] = E\{\mathbf{r} \cos[\omega n + \phi]\} = E\{\mathbf{r}\} E\{\cos[\omega n + \phi]\} = 0 \quad (5.221)$$

since  $\mathbf{r}$  and  $\phi$  are independent and

$$E\{\cos[\omega n + \phi]\} = \int_{-\pi}^{\pi} \cos[\omega n + \phi] \frac{1}{2\pi} d\phi = 0 \quad (5.222)$$

Its autocorrelation is given by

$$\begin{aligned} R_{xx}[n_1, n_2] &= E\{\mathbf{r}^2\} \int_{-\pi}^{\pi} \cos[\omega n_1 + \phi] \cos[\omega n_2 + \phi] \frac{1}{2\pi} d\phi \\ &= \frac{1}{2} E\{\mathbf{r}^2\} \int_{-\pi}^{\pi} \{\cos[\omega(n_1 + n_2) + \phi] + \cos[\omega(n_2 - n_1)]\} \frac{1}{2\pi} d\phi \\ &= \frac{1}{2} E\{\mathbf{r}^2\} \cos[\omega(n_2 - n_1)] \end{aligned} \quad (5.223)$$

which only depends on the time difference  $n_2 - n_1$ .

An important property of a stochastic process is that its autocorrelation  $R_{xx}[n_1, n_2]$  is a *positive-definite* function, i.e., for any  $a_i, a_j$

$$\sum_i \sum_j a_i a_j^* R_{xx}[n_i, n_j] \geq 0 \quad (5.224)$$

which is a consequence of the identity

$$0 \leq E \left\{ \left| \sum_i a_i \mathbf{x}[n_i] \right|^2 \right\} = \sum_i \sum_j a_i a_j^* E \{ \mathbf{x}[n_i] \mathbf{x}^*[n_j] \} \quad (5.225)$$

Similarly, the autocovariance of a complex random process is defined as

$$C_{xx}[n_1, n_2] = E \{ (\mathbf{x}[n_1] - \mu_x[n_1]) (\mathbf{x}[n_2] - \mu_x[n_2])^* \} = R_{xx}[n_1, n_2] - \mu_x[n_1] \mu_x^*[n_2] \quad (5.226)$$

The correlation coefficient of process  $\mathbf{x}[n]$  is defined as

$$r_{xx}[n_1, n_2] = \frac{C_{xx}[n_1, n_2]}{\sqrt{C_{xx}[n_1, n_1] C_{xx}[n_2, n_2]}} \quad (5.227)$$

An important property of the correlation coefficient is that it is bounded by 1:

$$|r_{xx}[n_1, n_2]| \leq 1 \quad (5.228)$$

which is the *Cauchy-Schwarz* inequality. To prove it, we note that for any real number  $a$

$$\begin{aligned} 0 &\leq E \left\{ \left| a(\mathbf{x}[n_1] - \mu[n_1]) + (\mathbf{x}[n_2] - \mu[n_2]) \right|^2 \right\} \\ &= a^2 C_{xx}[n_1, n_1] + 2a C_{xx}[n_1, n_2] + C_{xx}[n_2, n_2] \end{aligned} \quad (5.229)$$

Since the quadratic function in Eq. (5.229) is positive for all  $a$ , its roots have to be complex, and thus its discriminant has to be negative:

$$C_{xx}^2[n_1, n_2] - C_{xx}[n_1, n_1] C_{xx}[n_2, n_2] \leq 0 \quad (5.230)$$

from which Eq. (5.228) is derived.

The cross-correlation of two stochastic processes  $\mathbf{x}[n]$  and  $\mathbf{y}[n]$  is defined as

$$R_{xy}[n_1, n_2] = E \{ \mathbf{x}[n_1] \mathbf{y}^*[n_2] \} = R_{yx}^*[n_2, n_1] \quad (5.231)$$

where we have explicitly indicated with subindices the random process. Similarly, their cross-covariance is

$$C_{xy}[n_1, n_2] = R_{xy}[n_1, n_2] - \mu_x[n_1]\mu_y^*[n_2] \quad (5.232)$$

Two processes  $\mathbf{x}[n]$  and  $\mathbf{y}[n]$  are called *orthogonal* iff

$$R_{xy}[n_1, n_2] = 0 \quad \text{for every } n_1 \text{ and } n_2 \quad (5.233)$$

They are called *uncorrelated* iff

$$C_{xy}[n_1, n_2] = 0 \quad \text{for every } n_1 \text{ and } n_2 \quad (5.234)$$

*Independent processes.* If two processes  $\mathbf{x}[n]$  and  $\mathbf{y}[n]$  are such that the random variables  $\mathbf{x}[n_1], \mathbf{x}[n_2], \dots, \mathbf{x}[n_m]$ , and  $\mathbf{y}[n'_1], \mathbf{y}[n'_2], \dots, \mathbf{y}[n'_m]$  are mutually independent, then these processes are called independent. If two processes are independent, then they are also uncorrelated, though the converse is not generally true.

*Gaussian processes.* A process  $\mathbf{x}[n]$  is called Gaussian if the random variables  $\mathbf{x}[n_1], \mathbf{x}[n_2], \dots, \mathbf{x}[n_m]$  are jointly Gaussian for any  $m$  and  $n_1, n_2, \dots, n_m$ . If two processes are Gaussian and also uncorrelated, then they are also statistically independent.

### 5.8.2. Stationary Processes

Stationary processes are those whose statistical properties do not change over time. While truly stationary processes do not exist in speech signals, they are a reasonable approximation and have the advantage of allowing us to use the Fourier transforms defined in Section 5.1.3.3. In this section we define stationarity and analyze some of its properties.

A stochastic process is called *strict-sense stationary* (SSS) if its statistical properties are invariant to a shift of the origin: *i.e.*, both processes  $\mathbf{x}[n]$  and  $\mathbf{x}[n+l]$  have the same statistics for any  $l$ . Likewise, two processes  $\mathbf{x}[n]$  and  $\mathbf{y}[n]$  are called *jointly strict-sense stationary* if their joint statistics are the same as those of  $\mathbf{x}[n+l]$  and  $\mathbf{y}[n+l]$  for any  $l$ .

From the definition, it follows that the  $m^{\text{th}}$ -order density of an SSS process must be such that

$$f(x_1, \dots, x_m; n_1, \dots, n_m) = f(x_1, \dots, x_m; n_1 + l, \dots, n_m + l) \quad (5.235)$$

for any  $l$ . Thus the first-order density satisfies  $f(x, n) = f(x, n+l)$  for any  $l$ , which means that it is independent of  $n$ :

$$f(x, n) = f(x) \quad (5.236)$$

or, in other words, the density function is constant with time.

Similarly,  $f(x_1, x_2; n_1 + l, n_2 + l)$  is independent of  $l$ , which leads to the conclusion

$$f(x_1, x_2; n_1, n_2) = f(x_1, x_2; m) \quad m = n_1 - n_2 \quad (5.237)$$

or, in other words, the joint density of  $\mathbf{x}[n]$  and  $\mathbf{x}[n+m]$  is not a function of  $n$ , only of  $m$ , the time difference between the two samples.

Let's compute the first two moments of a SSS process:

$$E\{x[n]\} = \int x[n]f(x[n]) = \int xf(x) = \mu \quad (5.238)$$

$$E\{x[n+m]x^*[n]\} = \int x[n+m]x^*[n]f(x[n+m], x[n]) = R_{xx}[m] \quad (5.239)$$

or, in other words, its mean is not a function of time and its autocorrelation depends only on  $m$ .

A stochastic process  $\mathbf{x}[n]$  that obeys Eq. (5.238) and (5.239) is called *wide-sense stationary* (WSS). From this definition, a SSS process is also a WSS process but the converse is not true in general. Gaussian processes are an important exception, and it can be proved that a WSS Gaussian process is also SSS.

For example, the random process of Eq. (5.213) is WSS, because it has zero mean and its autocorrelation function, as given by Eq. (5.223), is only a function of  $m = n_1 - n_2$ . By setting  $m = 0$  in Eq. (5.239) we see that the average power of a WSS stationary process

$$E\{|x[n]|^2\} = R[0] \quad (5.240)$$

is independent of  $n$ .

The autocorrelation of a WSS process is a conjugate-symmetric function, also referred to as a *Hermitian* function:

$$R[-m] = E\{x[n-m]x^*[n]\} = E\{x[n]x^*[n+m]\} = R^*[m] \quad (5.241)$$

so that if  $x[n]$  is real,  $R[m]$  is even.

From Eqs. (5.219), (5.238), and (5.239) we can compute the autocovariance as

$$C[m] = R[m] - |\mu|^2 \quad (5.242)$$

and its correlation coefficient as

$$r[m] = C[m]/C[0] \quad (5.243)$$

Two processes  $\mathbf{x}[n]$  and  $\mathbf{y}[n]$  are called jointly WSS if both are WSS and their cross-correlation depends only on  $m = n_1 - n_2$ :

$$R_{xy}[m] = E\{x[n+m]y^*[n]\} \quad (5.244)$$

$$C_{xy}[m] = R_{xy}[m] - \mu_x \mu_y^* \quad (5.245)$$



### 5.8.2.1. Ergodic Processes

A critical problem in the theory of stochastic processes is the estimation of their various statistics, such as the mean and autocorrelation given that often only one realization of the random process is available. The first approximation would be to replace the expectation in Eq. (5.218) with its *ensemble average*:

$$\mu[n] \cong \frac{1}{M} \sum_{i=0}^{M-1} x_i[n] \quad (5.246)$$

where  $x_i[n]$  are different samples of the random process.

As an example, let  $\mathbf{x}[n]$  be the frequency-modulated (FM) random process received by a FM radio receiver:

$$\mathbf{x}[n] = a[n] + \mathbf{v}[n] \quad (5.247)$$

which contains some additive noise  $\mathbf{v}[n]$ . The realization  $x_i[n]$  received by receiver  $i$  will be different from the realization  $x_j[n]$  for receiver  $j$ . We know that each signal has a certain level of noise, so one would hope that by averaging them, we could get the mean of the process for a sufficiently large number of radio receivers.

In many cases, however, only one sample of the process is available. According to Eq. (5.246) this would mean that the sample signal equals the mean, which does not seem very robust. We could also compute the signal's time average, but this may not tell us much about the random process in general. However, for a special type of random processes called ergodic, their ensemble averages equal appropriate time averages.

A process  $\mathbf{x}[n]$  with constant mean

$$E\{\mathbf{x}[n]\} = \mu \quad (5.248)$$

is called *mean-ergodic* if, with probability 1, the ensemble average equals the time average when  $N$  approaches infinity:

$$\lim_{N \rightarrow \infty} \mu_N = \mu \quad (5.249)$$

where  $\mu_N$  is the time average

$$\mu_N = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} \mathbf{x}[n] \quad (5.250)$$

which, combined with Eq. (5.248), indicates that  $\mu_N$  is a random variable with mean  $\mu$ . Taking expectations in Eq. (5.250) and using Eq. (5.248), it is clear that

$$E\{\mu_N\} = \mu \quad (5.251)$$

so that proving Eq. (5.249) is equivalent to proving

$$\lim_{N \rightarrow \infty} \sigma_N^2 = 0 \quad (5.252)$$

with  $\sigma_N^2$  being the variance of  $\mu_N$ . It can be shown [12] that a process  $\mathbf{x}[n]$  is mean ergodic iff

$$\lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{n=-N/2}^{N/2-1} \sum_{m=-N/2}^{N/2-1} C_{xx}[n, m] = 0 \quad (5.253)$$

It can also be shown [12] that a sufficient condition for a WSS process to be mean ergodic is to satisfy

$$\lim_{m \rightarrow \infty} C_{xx}[m] = 0 \quad (5.254)$$

which means that if the random variables  $\mathbf{x}[n]$  and  $\mathbf{x}[n+m]$  are uncorrelated for large  $m$ , then process  $\mathbf{x}[n]$  is mean ergodic. This is true for many regular processes.

A similar condition can be proven for a WSS process to be covariance ergodic. In most cases in this book we assume ergodicity, first because of convenience for mathematical tractability, and second because it is a good approximation to assume that samples that are far apart are uncorrelated. *Ergodicity allows us to compute means and covariances of random processes by their time averages.*

### 5.8.3. LTI Systems with Stochastic Inputs

If the WSS random process  $\mathbf{x}[n]$  is the input to an LTI system with impulse response  $h[n]$ , the output

$$\mathbf{y}[n] = \sum_{m=-\infty}^{\infty} h[m]\mathbf{x}[n-m] = \sum_{m=-\infty}^{\infty} h[n-m]\mathbf{x}[m] \quad (5.255)$$

is another WSS random process. To prove this we need to show that the mean is not a function of  $n$ :

$$\mu_y[n] = E\{\mathbf{y}[n]\} = \sum_{m=-\infty}^{\infty} h[m]E\{\mathbf{x}[n-m]\} = \mu_x \sum_{m=-\infty}^{\infty} h[m] \quad (5.256)$$

The cross-correlation between input and output is given by

$$\begin{aligned} R_{xy}[m] &= E\{\mathbf{x}[n+m]\mathbf{y}[n]\} = \sum_{l=-\infty}^{\infty} h^*[l]E\{\mathbf{x}[n+m]\mathbf{x}[n-l]\} \\ &= \sum_{l=-\infty}^{\infty} h^*[l]R_{xx}[m+l] = \sum_{l=-\infty}^{\infty} h^*[-l]R_{xx}[m-l] = h^*[-m] * R_{xx}[m] \end{aligned} \quad (5.257)$$

and the autocorrelation of the output

$$\begin{aligned}
R_{yy}[m] &= E\{y[n+m]y^*[n]\} = \sum_{l=-\infty}^{\infty} h[l]E\{x[n+m-l]y^*[n]\} \\
&= \sum_{l=-\infty}^{\infty} h[l]R_{xy}[m-l] = h[m] * R_{xy}[m] = h[m] * h^*[-m] * R_{xx}[m]
\end{aligned} \tag{5.258}$$

is only a function of  $m$ .

#### 5.8.4. Power Spectral Density

The Fourier transform of a WSS random process  $\mathbf{x}[n]$  is a stochastic process in the variable  $\omega$

$$\mathbf{X}(\omega) = \sum_{n=-\infty}^{\infty} \mathbf{x}[n]e^{-j\omega n} \tag{5.259}$$

whose autocorrelation is given by

$$\begin{aligned}
E\{\mathbf{X}(\omega+u)\mathbf{X}^*(\omega)\} &= E\left\{\sum_{l=-\infty}^{\infty} \mathbf{x}[l]e^{-j(\omega+u)l} \sum_{m=-\infty}^{\infty} \mathbf{x}^*[m]e^{j\omega m}\right\} = \\
&= \sum_{n=-\infty}^{\infty} e^{-j(\omega+u)n} \sum_{m=-\infty}^{\infty} E\{\mathbf{x}[m+n]\mathbf{x}^*[m]\}e^{-jum}
\end{aligned} \tag{5.260}$$

where we made a change of variables  $l = n + m$  and changed the order of expectation and summation. Now, if  $\mathbf{x}[n]$  is WSS

$$R_{xx}[n] = E\{\mathbf{x}[m+n]\mathbf{x}^*[m]\} \tag{5.261}$$

and if we set  $u = 0$  in Eq. (5.260) together with Eq. (5.261), then we obtain

$$S_{xx}(\omega) = E\{|\mathbf{X}(\omega)|^2\} = \sum_{n=-\infty}^{\infty} R_{xx}[n]e^{-j\omega n} \tag{5.262}$$

$S_{xx}(\omega)$  is called the *power spectral density* of the WSS random process  $\mathbf{x}[n]$ , and it is the Fourier transform of its autocorrelation function  $R_{xx}[n]$ , with the inversion formula being

$$R_{xx}[n] = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(\omega)e^{j\omega n} d\omega \tag{5.263}$$

Note that Eqs. (5.48) and (5.263) are identical, though in one case we compute the autocorrelation of a signal as a time average, and the other is the autocorrelation of a random process as an ensemble average. For an ergodic process both are the same.

Just as we take Fourier transforms of deterministic signals, we can also compute the power spectral density of a random process as long as it is wide-sense stationary, which is why these wide-sense stationary processes are so useful.

If the random process  $\mathbf{x}[n]$  is real then  $R_{xx}[n]$  is real and even and, using properties in Table 5.5,  $S_{xx}(\omega)$  is also real and even.

Parseval's theorem for random processes also applies here:

$$E\{|\mathbf{x}[n]|^2\} = R_{xx}[0] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{xx}(\omega) d\omega \quad (5.264)$$

so that we can compute the signal's energy from the area under  $S_{xx}(\omega)$ . Let's get a physical interpretation of  $S_{xx}(\omega)$ . In order to do that we can similarly derive the cross-power spectrum  $S_{xy}(\omega)$  of two WSS random processes  $\mathbf{x}[n]$  and  $\mathbf{y}[n]$  as the Fourier transform of their cross-correlation:

$$S_{xy}(\omega) = \sum_{n=-\infty}^{\infty} R_{xy}[n] e^{-j\omega n} \quad (5.265)$$

which allows us, taking Fourier transforms in Eq. (5.257), to obtain the cross-power spectrum between input and output to a linear system as

$$S_{xy}(\omega) = S_{xx}(\omega) H^*(\omega) \quad (5.266)$$

Now, taking the Fourier transform of Eq. (5.258), the power spectrum of the output is thus given by

$$S_{yy}(\omega) = S_{xy}(\omega) H(\omega) = S_{xx}(\omega) |H(\omega)|^2 \quad (5.267)$$

Finally, suppose we filter  $\mathbf{x}[n]$  through the ideal bandpass filter

$$H_b(\omega) = \begin{cases} \sqrt{\pi/c} & \omega_0 - c < \omega < \omega_0 + c \\ 0 & \text{otherwise} \end{cases} \quad (5.268)$$

The energy of the output process is

$$0 \leq E\{|\mathbf{y}[n]|^2\} = R_{yy}[0] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{yy}(\omega) d\omega = \frac{1}{2c} \int_{\omega_0-c}^{\omega_0+c} S_{xx}(\omega) d\omega \quad (5.269)$$

so that taking the limit when  $c \rightarrow 0$  results in

$$0 \leq \lim_{c \rightarrow 0} \frac{1}{2c} \int_{\omega_0-c}^{\omega_0+c} S_{xx}(\omega) d\omega = S_{xx}(\omega_0) \quad (5.270)$$

which is the *Wiener-Khinchin* theorem and says that the power spectrum of a WSS process  $\mathbf{x}[n]$ , real or complex, is always positive for any  $\omega$ . Equation (5.269) also explains the

name power spectral density, because  $S_{xx}(\omega)$  represents the density of power at any given frequency  $\omega$ .

### 5.8.5. Noise

A process  $\mathbf{x}[n]$  is *white noise* if, and only if, its samples are uncorrelated:

$$C_{xx}[n_1, n_2] = C[n_1]\delta[n_1 - n_2] \quad (5.271)$$

and is zero-mean  $\mu_x[n] = 0$ .

If in addition  $\mathbf{x}[n]$  is WSS, then

$$C_{xx}[n] = R_{xx}[n] = q\delta[n] \quad (5.272)$$

which has a flat power spectral density

$$S_{xx}(\omega) = q \quad \text{for all } \omega \quad (5.273)$$

The thermal noise phenomenon in metallic resistors can be accurately modeled as white Gaussian noise. White noise doesn't have to be Gaussian (white Poisson impulse noise is one of many other possibilities).

*Colored noise* is defined as a zero-mean WSS process whose samples are correlated with autocorrelation  $R_{xx}[n]$ . Colored noise can be generated by passing white noise through a filter  $h[n]$  such that  $S_{xx}(\omega) = |H(\omega)|^2$ . A type of colored noise that is very frequently encountered in speech signals is the so-called *pink noise*, whose power spectral density decays with  $\omega$ . A more in-depth discussion of noise and its effect on speech signals is included in Chapter 10.

## 5.9. HISTORICAL PERSPECTIVE AND FURTHER READING

It is impossible to cover the field of Digital Signal Processing in just one chapter. The book by Oppenheim and Shafer [10] is one of the most widely used as a comprehensive treatment. For a more in-depth coverage of digital filter design, you can read the book by Parks and Burrus [13]. A detailed study of the FFT is provided by Burrus and Parks [2]. The theory of signal processing for analog signals can be found in Oppenheim and Willsky [11]. The theory of random signals can be found in Papoulis [12]. Multirate processing is well studied in Crochiere and Rabiner [4]. Razavi [16] covers analog-digital conversion. Software programs, such as MATLAB [1], contain a large number of packaged subroutines. Malvar [7] has extensive coverage of filterbanks and lapped transforms.

The field of Digital Signal Processing has a long history. The greatest advances in the field started in the 17<sup>th</sup> century. In 1666, English mathematician and physicist Sir *Isaac Newton* (1642-1727) invented differential and integral calculus, which was independently

discovered in 1675 by German mathematician *Gottfried Wilhelm Leibniz* (1646-1716). They both developed discrete mathematics and numerical methods to solve such equations when closed-form solutions were not available. In the 18<sup>th</sup> century, these techniques were further extended. Swiss brothers *Johann* (1667-1748) and *Jakob Bernoulli* (1654-1705) invented the calculus of variations and polar coordinates. French mathematician *Joseph Louis Lagrange* (1736-1813) developed algorithms for numerical integration and interpolation of continuous functions. The famous Swiss mathematician *Leonhard Euler* (1707-1783) developed the theory of complex numbers and number theory so useful in the DSP field, in addition to the first full analytical treatment of algebra, the theory of equations, trigonometry and analytical geometry. In 1748, Euler examined the motion of a vibrating string and discovered that sinusoids are eigenfunctions for linear systems. Swiss scientist *Daniel Bernoulli* (1700-1782), son of Johann Bernoulli, also conjectured in 1753 that all physical motions of a string could be represented by linear combinations of normal modes. However, both Euler and Bernoulli, and later Lagrange, discarded the use of trigonometric series because *it was impossible to represent signals with corners*. The 19<sup>th</sup> century brought us the theory of harmonic analysis. One of those who contributed most to the field of Digital Signal Processing is *Jean Baptiste Joseph Fourier* (1768-1830), a French mathematician who in 1822 published *The Analytical Theory of Heat*, where he derived a mathematical formulation for the phenomenon of heat conduction. In this treatise, he also developed the concept of Fourier series and harmonic analysis and the Fourier transform. One of Fourier's disciples, the French mathematician *Simeon-Denis Poisson* (1781-1840), studied the convergence of Fourier series together with countryman *Augustin Louis Cauchy* (1789-1857). Nonetheless, it was German *Peter Dirichlet* (1805-1859) who gave the first set of conditions sufficient to guarantee the convergence of a Fourier series. French mathematician *Pierre Simon Laplace* (1749-1827) invented the Laplace transform, a transform for continuous-time signals over the whole complex plane. French mathematician *Marc-Antoine Parseval* (1755-1836) derived the theorem that carries his name. German *Leopold Kronecker* (1823-1891) did work with discrete delta functions. French mathematician *Charles Hermite* (1822-1901) discovered complex conjugate matrices. American *Josiah Willard Gibbs* (1839-1903) studied the phenomenon of Fourier approximations to periodic square waveforms.

Until the early 1950s, all signal processing was analog, including the *long-playing* (LP) record first released in 1948. Pulse Code Modulation (PCM) had been invented by *Paul M. Rainey* in 1926 and independently by *Alan H. Reeves* in 1937, but it wasn't until 1948 when *Oliver, Pierce, and Shannon* [9] laid the groundwork for PCM (see Chapter 7 for details). Bell Labs engineers developed a PCM system in 1955, the so-called T-1 carrier system, which was put into service in 1962 as the world's first common-carrier digital communications system and is still used today. The year 1948 also saw the invention of the transistor at Bell Labs and a small prototype computer at Manchester University and marked the birth of modern Digital Signal Processing. In 1958, *Jack Kilby* of Texas Instruments invented the integrated circuit and in 1970, researchers at Lincoln Laboratories developed the first real-time DSP computer, which performed signal processing tasks about 100 times faster than general-purpose computers of the time. In 1978, Texas Instruments introduced *Speak & Spell*<sup>TM</sup>, a toy that included an integrated circuit especially designed for speech synthesis. Intel Corporation introduced in 1971 the 4-bit Intel 4004, the first general-purpose

microprocessor chip, and in 1972 they introduced the 8-bit 8008. In 1982 Texas Instruments introduced the TMS32010, the first commercially viable single-chip Digital Signal Processor (DSP), a microprocessor specially designed for fast signal processing operations. At a cost of about \$100, the TMS32010 was a 16-bit fixed-point chip with a hardware multiplier built-in that executed 5 million instructions per second (MIPS). Gordon Moore, Intel's founder, came up with the law that carries his name stating that computing power doubles every 18 months, allowing ever faster processors. By the end of the 20<sup>th</sup> century, DSP chips could perform floating-point operations at a rate over 1000MIPS and had a cost below \$5, so that today they are found in many devices from automobiles to cellular phones.

While hardware improvements significantly enabled the development of the field, digital algorithms were also needed. The 1960s saw the discovery of many of the concepts described in this chapter. In 1965, *James W. Cooley* and *John W. Tukey* [3] discovered the FFT, although it was later found [6] that German mathematician *Carl Friedrich Gauss* (1777-1855) had already invented it over a century earlier. The FFT sped up calculations by orders of magnitude, which opened up many possible algorithms for the slow computers of the time. *James F. Kaiser*, *Bernard Gold*, and *Charles Rader* published key papers on digital filtering. *John Stockham* and *Howard Helms* independently discovered fast convolution by doing convolution with FFTs.

An association that has had a large impact on the development of modern Digital Signal Processing is the Institute of Electrical and Electronic Engineers (IEEE), which has over 350,000 members in 150 nations and is the world's largest technical organization. It was founded in 1884 as the American Institute of Electrical Engineers (AIEE). IEEE's other parent organization, the Institute of Radio Engineers (IRE), was founded in 1912, and the two merged in 1963. The IEEE Signal Processing Society is a society within the IEEE devoted to Signal Processing. Originally founded on 1948 as the Institute of Radio Engineers Professional Group on Audio, it was later renamed the IEEE Group on Audio (1964), the IEEE Audio and Electroacoustics group (1965), the IEEE group on Acoustics Speech and Signal Processing (1974), the Acoustic, Speech and Signal Processing Society (1976), and finally IEEE Signal Processing Society (1989). In 1976 the society initiated its practice of holding an annual conference, the International Conference on Acoustic, Speech and Signal Processing (ICASSP), which has been held every year since, and whose proceedings constitute an invaluable reference. *Frederik Nebeker* [8] provides a history of the society's first 50 years rich in insights from the pioneers.

## REFERENCES

- [1] Burrus, C.S., *et al.*, *Computer-Based Exercises for Signal Processing Using Matlab*, 1994, Upper Saddle River, NJ, Prentice Hall.
- [2] Burrus, C.S. and T.W. Parks, *DFT/FFT and Convolution Algorithms: Theory and Implementation*, 1985, New York, John Wiley.
- [3] Cooley, J.W. and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, 1965, **19**(Apr.), pp. 297-301.
- [4] Crochiere, R.E. and L.R. Rabiner, *Multirate Digital Signal Processing*, 1983, Upper Saddle River, NJ, Prentice-Hall.

- [5] Duhamel, P. and H. Hollman, "Split Radix FFT Algorithm," *Electronic Letters*, 1984, **20**(January), pp. 14-16.
- [6] Heideman, M.T., D.H. Johnson, and C.S. Burrus, "Gauss and the History of the Fast Fourier Transform," *IEEE ASSP Magazine*, 1984, **1**(Oct), pp. pp 14-21.
- [7] Malvar, H., *Signal Processing with Lapped Transforms*, 1992, Artech House.
- [8] Nebeker, F., *Fifty Years of Signal Processing: The IEEE Signal Processing Society and its Technologies*, 1998, IEEE.
- [9] Oliver, B.M., J.R. Pierce, and C. Shannon, "The Philosophy of PCM," *Proc. Institute of Radio Engineers*, 1948, **36**, pp. pp 1324-1331.
- [10] Oppenheim, A.V. and R.W. Schaffer, *Discrete-Time Signal Processing*, 1999, Prentice-Hall, Upper Saddle River, NJ.
- [11] Oppenheim, A.V. and A.S. Willsky, *Signals and Systems*, 1997, Upper Saddle River, NJ, Prentice-Hall.
- [12] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, 3<sup>rd</sup> ed, 1991, New York, McGraw-Hill.
- [13] Parks, T.W. and C.S. Burrus, *Digital Filter Design*, 1987, New York, John Wiley.
- [14] Parks, T.W. and J.H. McClellan, "A Program for the Design of Linear Phase Finite Impulse Response Filters," *IEEE Trans. on Audio Electroacoustics*, 1972, **AU-20**(Aug), pp. 195-199.
- [15] Rao, K.R. and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages and Applications*, 1990, San Diego, CA, Academic Press.
- [16] Razavi, B., *Principles of Data Conversion System Design*, 1995, IEEE Press.
- [17] Smith, M.J.T. and T.P. Barnwell, "A Procedure for Designing Exact Reconstruction Filter Banks for Tree Structured Subband Coders," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1984, San Diego, Calif pp. 27.1.1-27.1.4.